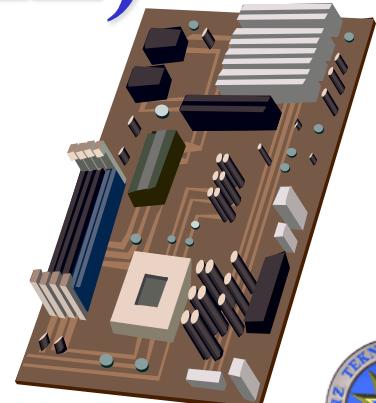




Makine Öğrenmesine Giriş (Machine Learning – ML)

Prof.Dr.Banu Diri

YILDIZ TEKNİK ÜNİVERSİTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ



Doğal Dil İşlemede Eğilimler

Önce : Yapay Zeka Tabanlı, Tam olarak anlama

Şimdi : Külliyat(Corpus)-tabanlı, İstatistikî, Makine Öğrenmesi Kullanan

Kanıt

1991 → Corpus tabanlı 3 makale

1996 → Bu alandaki makalelerin yarıdan fazlası Corpus tabanlı



AKİŞ

- Makine Öğrenmesi Nedir ?
- Günlük Hayatımızdaki Uygulamaları
- Verilerin Sayısallaştırılması
- Özellik Belirleme
 - Özellik Seçim Metotları
 - Bilgi Kazancı (Information Gain-IG)
 - Sinyal-Gürültüye Oranı: (S/N ratio)
 - Alt küme seçiciler (Wrappers)
 - Yeni Özelliklerin Çıkarımı
 - Temel Bileşen Analizi (Principal Component Analysis)
 - Doğrusal Ayırteden Analizi (Linear Discriminant Analysis)
- Sınıflandırma Metotları
 - Doğrusal Regresyon
 - SVM (Support Vector Machine)
 - Karar Ağaçları (Decision Trees)
 - Yapay Sinir Ağları
 - En Yakın K Komşu Algoritması (k - Nearest Neighbor)
 - Öğrenmeli Vektör Kuantalama (Learning Vector Quantization)
- Kümeleme Algoritmaları
 - K-Ortalama (K-Means)
 - Kendi Kendini Düzenleyen Haritalar (Self Organizing Map -SOM)

Makine Öğrenmesi

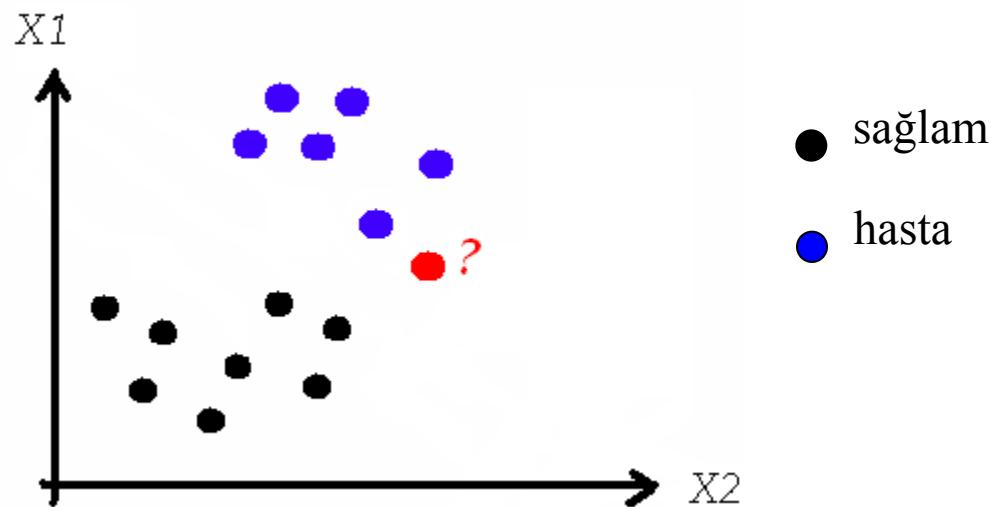
- Çok büyük miktarlardaki verinin elle işlenmesi ve analizinin yapılması mümkün değildir.
 - Amaç geçmişteki verileri kullanarak gelecek için tahminlerde bulunmaktır.
 - Bu problemleri çözmek için **Makine Öğrenmesi** (machine learning) yöntemleri geliştirilmiştir.
-
- Makine öğrenmesi yöntemleri, geçmişteki veriyi kullanarak yeni veri için en uygun modeli bulmaya çalışır.
 - Verinin incelenip, içerisindeki işe yarayan bilginin çıkarılmasına da **Veri Madenciliği** (data mining) adı verilir.

Metot türleri

- Farklı uygulamaların, analizlerden farklı beklenileri olmaktadır.
- Makine öğrenmesi metodlarını bu beklenilere göre sınıflandırmak mümkündür.

Sınıflandırma

Geçmiş bilgileri hangi sınıfından olduğu biliniyorsa, yeni gelen verinin hangi sınıfı dahil olacağının bulunmasıdır.

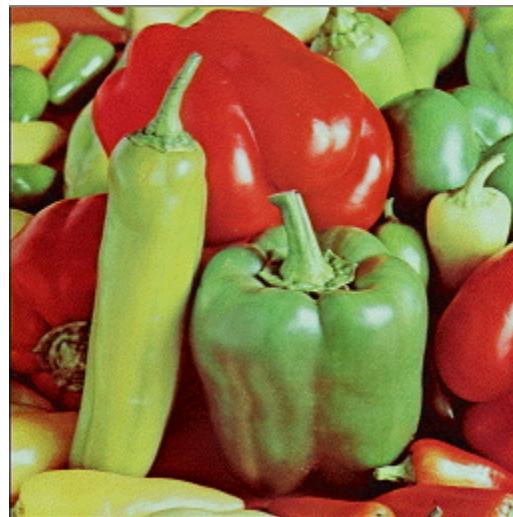


Kırmızı hangi sınıfı dahildir ?

Kümeleme

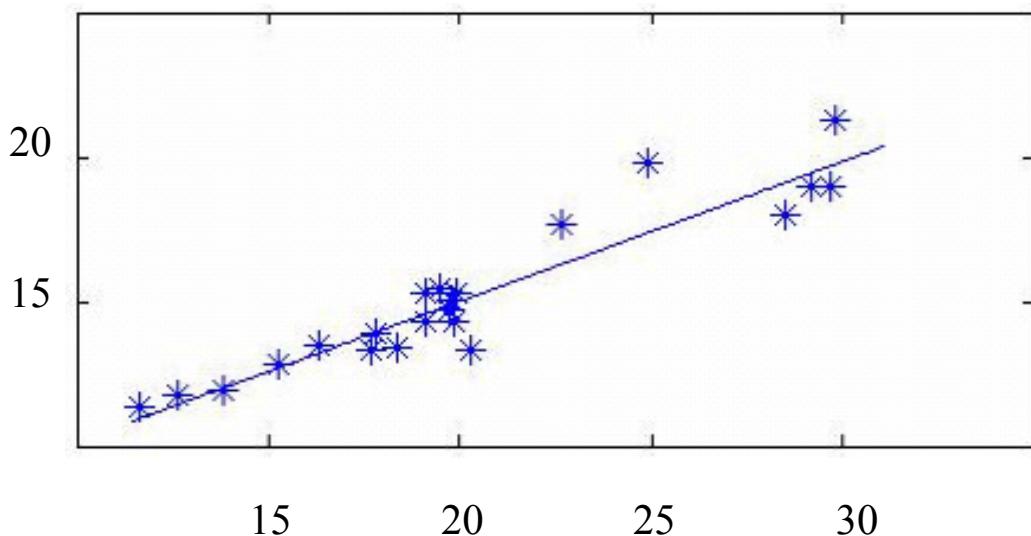
Geçmişteki verilerin sınıfları/etiketleri verilmemiği/bilinmediği durumlarda verilerin birbirlerine yakın benzerliklerinin yer aldığı kümelerin bulunmasıdır.

- 256 rengi 16 renge nasıl indiririz ?



Regresyon - Eğri Uydurma

Geçmiş bilgilere ait sınıflar yerine sürekli bilginin yer aldığı problemlerdir.



x ekseni hava sıcaklığını, y ekseni de deniz suyu sıcaklığını göstermektedir.

Bizden istenen hava sıcaklığına bağlı olarak deniz suyu sıcaklığının tahmin edilmesidir.

Giriş ile çıkış arasındaki fonksiyonun eğrisi bulunur.

Birliktelik Kuralları Keşfi

Bir süpermarkette, x ürününü alan müşterilerin %80'i y ürününü de alıyorsa, x ürününü alıp, y ürününü almayan müşteriler, y ürününün potansiyel müşterileridir. Müşterilerin sepet bilgilerinin bulunduğu bir veri tabanında potansiyel y müşterilerini bulma işlemi türündeki problemler ilişki belirme yöntemleri ile çözülür.

- Sepet analizi
- Raf düzenlemesi
- Promosyonlar



Özellik Seçimi

Veriye ait olan birçok özellikten bazıları ilgili verinin kümesini/sınıfinı belirlemede önemli rol oynar. Bu gibi durumlarda özellik kümesinin bir altkümesi seçilir (*özellik seçimi*) veya bu özelliklerin birleşiminden yeni özellikler elde edilebilir (*özellik çıkarımı*).

Kısaca, Makineler, isanlığın işgücüne sağladıkları yararı, makine öğrenmesi yöntemleri ile birleştiriklerinde beyin gücünü de sağlamayı başarmışlardır.

Uygulama alanı ne olursa olsun, çok miktardaki verinin analiz edilerek gelecek ile ilgili tahminlerde bulunması ve bizim karar vermemize yardımcı olması ile makine öğrenmesi yöntemlerinin her geçen gün önemi artmaktadır.

Akış

- Makine Öğrenmesi Nedir?
- **Günlük Hayatımızdaki Uygulamaları**
- Verilerin Sayısallaştırılması
- Özellik Belirleme
 - Özellik Seçim Metotları
 - Bilgi Kazancı (Information Gain-IG)
 - Sinyal-Gürültüye Oranı: (S/N ratio)
 - Alt küme seçiciler (Wrappers)
 - Yeni Özelliklerin Çıkarımı
 - Temel Bileşen Analizi (Principal Component Analysis)
 - Doğrusal Ayırteden Analizi (Linear Discriminant Analysis)
- Sınıflandırma Metotları
 - Doğrusal Regresyon
 - SVM (Support Vector Machine)
 - Karar Ağaçları (Decision Trees)
 - Yapay Sinir Ağları
 - En Yakın K Komşu Algoritması (k - Nearest Neighbor)
 - Öğrenmeli Vektör Kuantalama (Learning Vector Quantization)
- Kümeleme Algoritmaları
 - K-Ortalama (K-Means)
 - Kendi Kendini Düzenleyen Haritalar (Self Organizing Map -SOM)

Makine Öğrenmesinin Günlük Hayatımızdaki Uygulamaları

YILDIZ TEKNİK ÜNİVERSİTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ



El yazısı / Kitap Yazısı Tanıma HCR /OCR

Thanks a million for your support
of the Ellen MacArthur Trust.
I hope our biggest reward
will be the smile and laughter
of the kids out on the water!
lets go for it! 

and we measure the effect of three choices. It finds that one can get different implementation projects per implementation of the "Implementation". The results are as follows:

In language *abc* the following appear quite late (i.e., around problem-type 1000):

- value*: for instance *abc* is *constant*, positive & *quadratic* with some initial value.
- value*: for instance *abc* is *constant*, positive & *quadratic* with some initial value, it ends at 0, the constant positive quadratic part is zero, and the quadratic part is non-zero.
- value*: for instance *abc* is *constant*, positive & *quadratic* with some initial value, it ends at 0, the constant positive quadratic part is zero, and the quadratic part is zero.

After problem-type 1000, the following appear:

- value*: for instance *abc* is *constant*, positive & *quadratic* with some initial value, it ends at 0, the constant positive quadratic part is zero, and the quadratic part is non-zero.
- value*: for instance *abc* is *constant*, positive & *quadratic* with some initial value, it ends at 0, the constant positive quadratic part is zero, and the quadratic part is zero.

After problem-type 1000, the following appear:

- value*: for instance *abc* is *constant*, positive & *quadratic* with some initial value, it ends at 0, the constant positive quadratic part is zero, and the quadratic part is non-zero.
- value*: for instance *abc* is *constant*, positive & *quadratic* with some initial value, it ends at 0, the constant positive quadratic part is zero, and the quadratic part is zero.

After problem-type 1000, the following appear:

- value*: for instance *abc* is *constant*, positive & *quadratic* with some initial value, it ends at 0, the constant positive quadratic part is zero, and the quadratic part is non-zero.
- value*: for instance *abc* is *constant*, positive & *quadratic* with some initial value, it ends at 0, the constant positive quadratic part is zero, and the quadratic part is zero.

İşlem: Şekillerin hangi harf olduğunu tahmin etme



Kredi Taleplerini Değerlendirme

- Birisi bankadan kredi ister.
- Banka krediyi versin mi/vermesin mi ?
- Nasıl?



e-ticaret

- Birisi Amazon.com dan bir kitap ya da ürün alıyor.

Görev ne olabilir?



e-ticaret

- Birisi Amazon.com dan bir kitap yada ürün alıyor.

Görev ne olabilir?

Müşteriye alması muhtemel kitaplar önerilir.

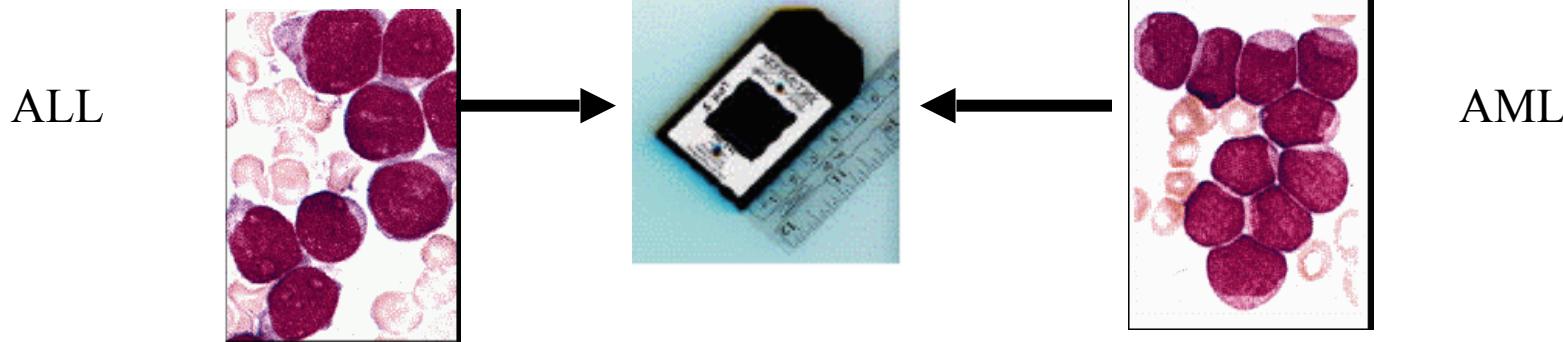
Ama nasıl?

Kitapları

- konularına
- yazarlarına
- birlikte satışlarına

göre kümelemek.





Gen Mikrodizilimleri

100 kişinin (hasta/sağlam) elimizde gen dizilimleri var.
 Bu dizilimleri analiz ederek hasta olup olmadığı
 bilinmeyen birisinin hasta olup olmadığını ya da
 hastalığının türünü öğrenebilir miyiz ?

En iyi tedaviyi öneremiz miyiz ?

Nasıl ? Elimizde hangi bilgiler olmalı ?

Bu adam kim? İçeri girsin mi?



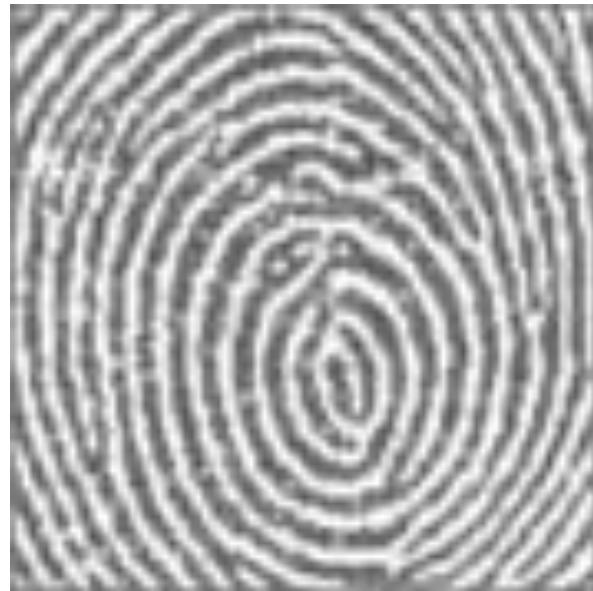
Bu adam havaalanında mı?



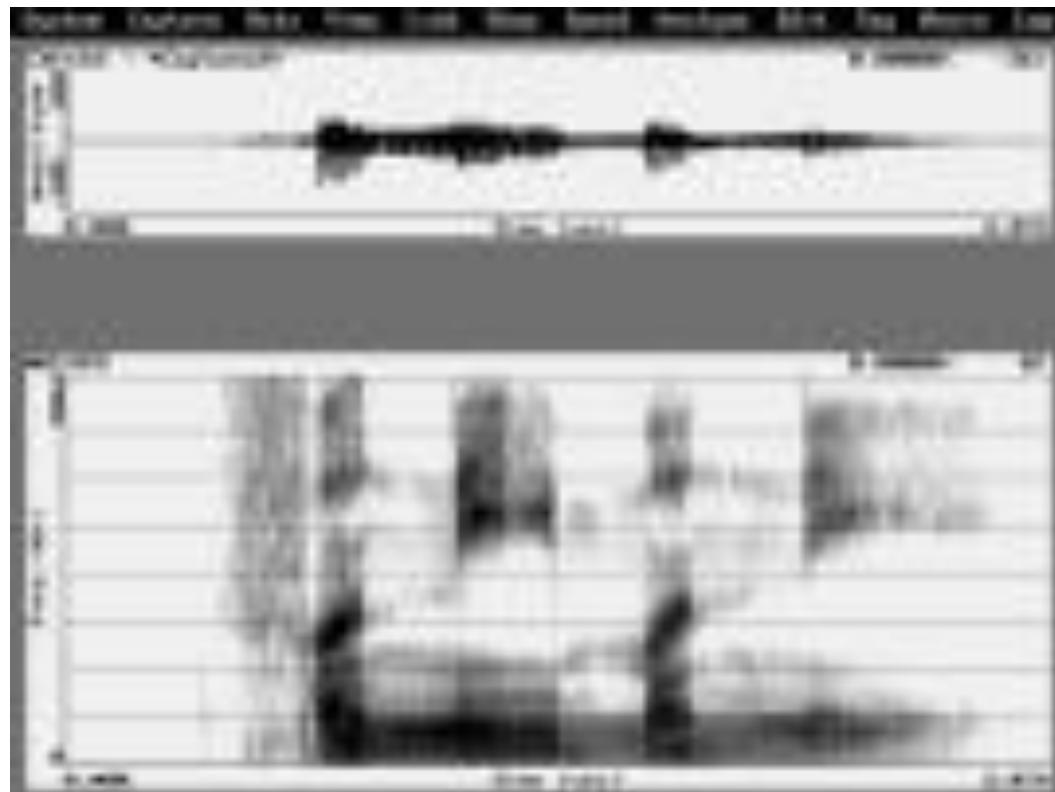
YILDIZ TEKNİK ÜNİVERSİTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ



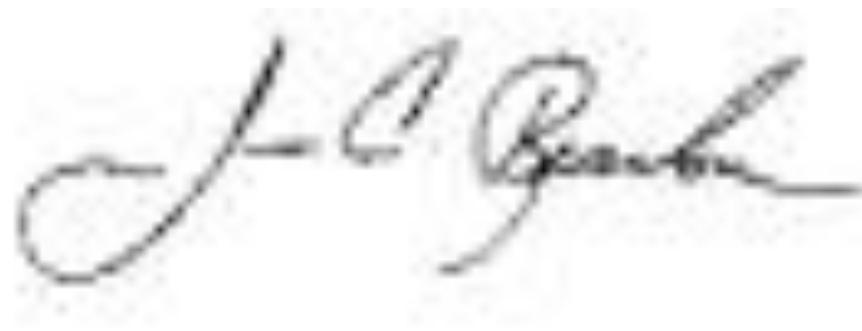
Bu parmak izi kimin ?
Bu adamı tutuklayalım mı?



Bu ses kimin ?
Bu ses ne diyor ?



Bu imza kimin ? Yoksa taklit mi?



Taklit olup olmadığını nasıl anlarız ?
Zaman bilgisi ?

Bu metnin konusu nedir?

Bu mail spam mi?



Anti spam yazılımları nasıl çalışır ?

Spam'ciler nasıl çalışıyor ?

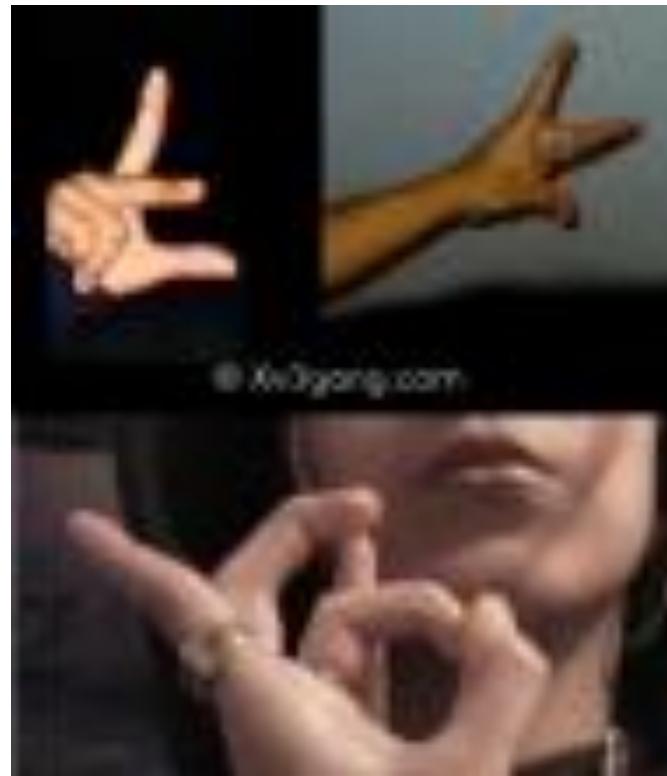
Yeni nesil spam mailler: Mesaj resimde,
metinde ise anti spamlardan kaçmak için gereken kelimeler var.

Makine öğrenmesi metodlarını hem spmciler hem anti spmciler kullanıyor.

Olağan dışı bir durum var mı ? Güvenlik kamerası kayıtları



Kamera kaydındaki kişi ne anlatıyor?



Sonuç: İletişimin artması.

YILDIZ TEKNİK ÜNİVERSİTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

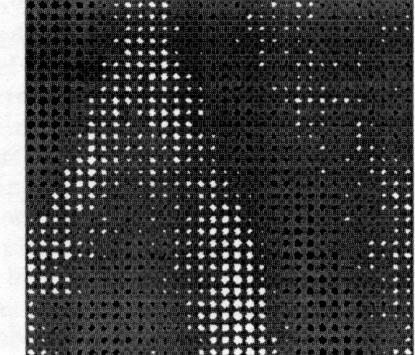
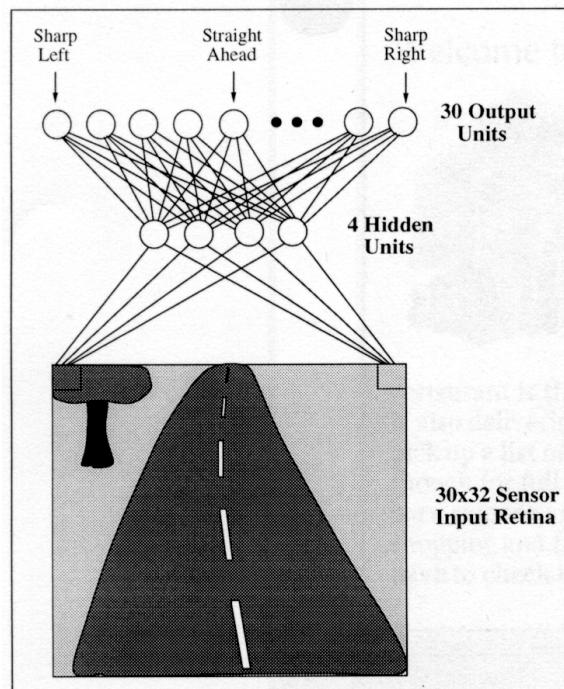
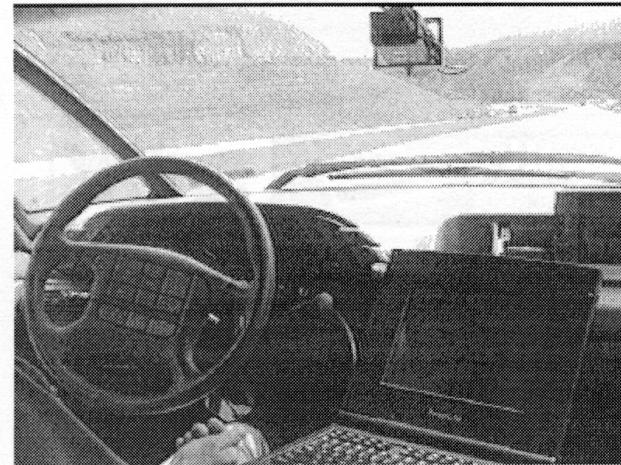


ALVIN

Otobanda saatte 70 mil hızla
sürücüsüz gidebilen bir
otomobil

Bütün denemeler trafige kapalı
alanlarda gerçekleştirılmıştır ☺

Neden şehiriçi değil ?
Neden otoban ?
Neden diğer arabalar yok ?
Araba birine çarparsa suçlu kim ?



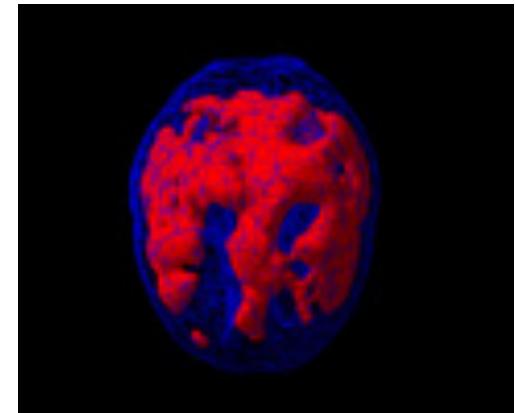
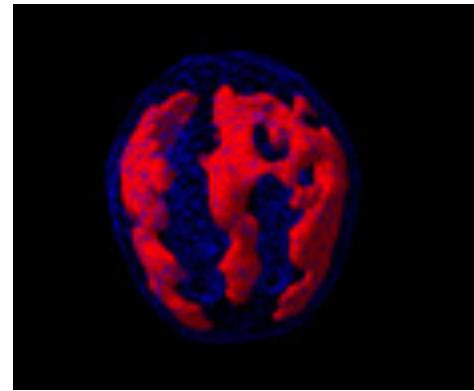
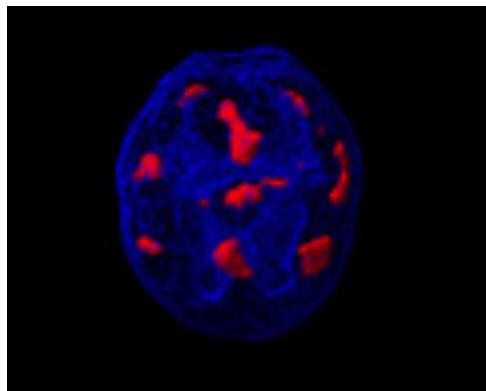
Adalet

- Çin'de pilot uygulama
 - Bir şehrin mahkeme hakimleri bir bilgisayar programı
 - Amaç: Daha adil bir dünya
 - Aynı özelliklere sahip davalarda aynı kararların alınması
 - Sistemin eğitimi için neler gerekli ?
 - Milyonlarca/Milyarlarca (buranın Çin olduğunu unutmayalım) davaya ait verilerin kategorilenmesi

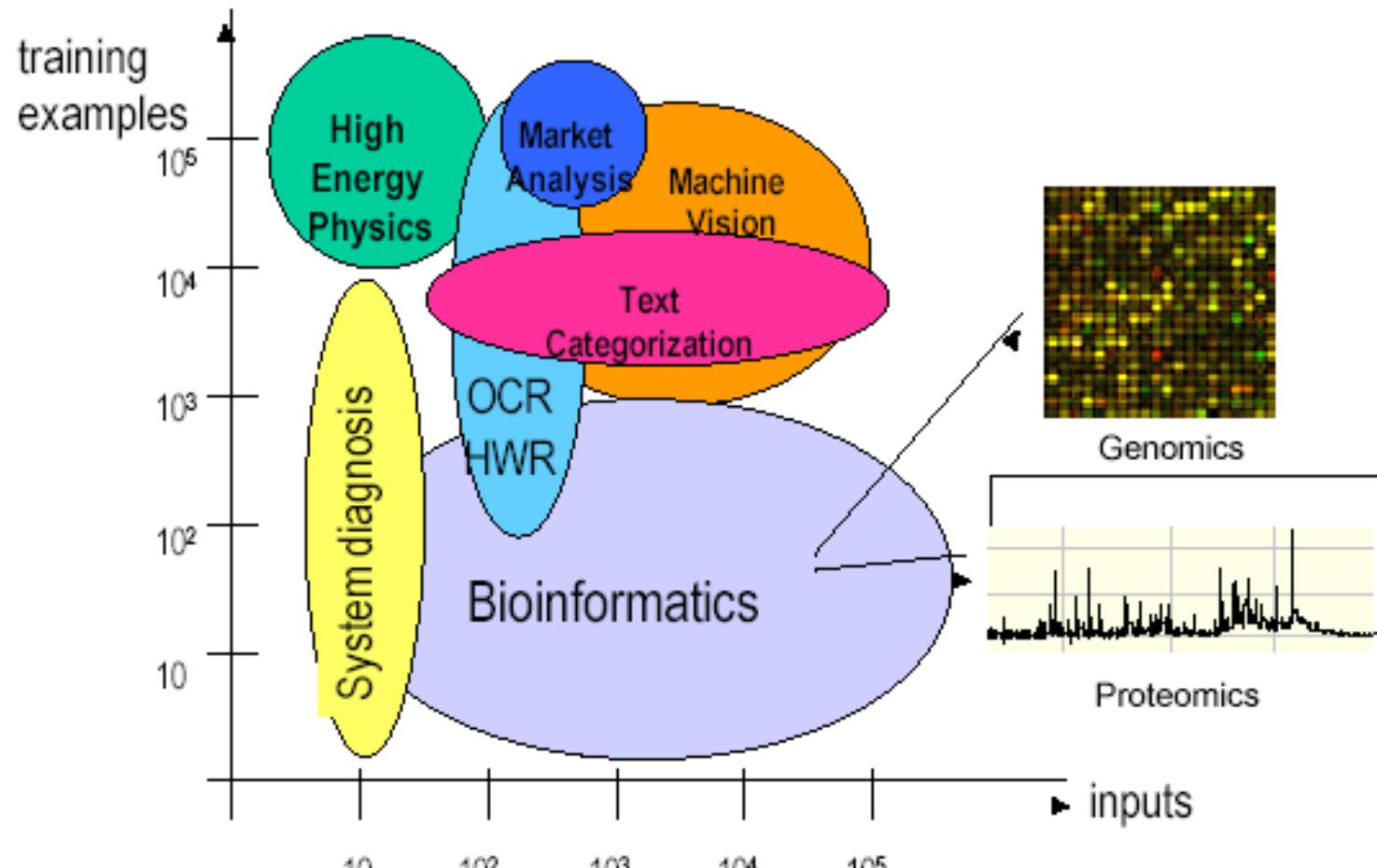


Beyin Aktiviteleri

- İnsanların
 - değişik şeyler düşünürken ki,
 - değişik duygulara sahipken ki,
 - problem çözerken ki
 - beyin aktiviteleri kaydedilir.
- Görev ?



Uygulamalardaki boyut örnek sayıları



AKİŞ

- Makine Öğrenmesi Nedir ?
- Günlük Hayatımızdaki Uygulamaları
- **Verilerin Sayısallaştırılması**
- Özellik Belirleme
 - Özellik Seçim Metotları
 - Bilgi Kazancı (Information Gain-IG)
 - Sinyal-Gürültü Oranı: (S/N ratio)
 - Alt küme seçiciler (Wrappers)
 - Yeni Özelliklerin Çıkarımı
 - Temel Bileşen Analizi (Principal Component Analysis)
 - Doğrusal Ayırteden Analizi (Linear Discriminant Analysis)
- Sınıflandırma Metotları
 - Doğrusal Regresyon
 - SVM (Support Vector Machine)
 - Karar Ağaçları (Decision Trees)
 - Yapay Sinir Ağları
 - En Yakın K Komşu Algoritması (k - Nearest Neighbor)
 - Öğrenmeli Vektör Kuantalama (Learning Vector Quantization)
- Kümeleme Algoritmaları
 - K-Ortalama (K-Means)
 - Kendi Kendini Düzenleyen Haritalar (Self Organizing Map -SOM)

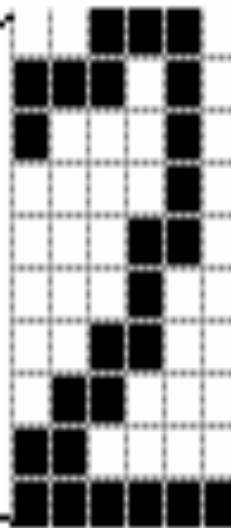
Verilerin Sayısallaştırılması

Makine öğrenmesi yöntemlerinin çoğunda işlenecek olan veri öncelikle sayılaştırılır.

Resim	Resmin her bir pikseli, renkli resimlerde R,G,B değerleri, siyah-beyaz resimlerde 1–255 arası gri seviyesi kullanılarak sayılara çevrilir. Renkli resimler 3 adet, siyah beyazlar 1 adet en*boy büyülüğünde matrisle ifade edilir.
Metin	Metindeki harfler, heceler ve kelimeler genelde frekanslarına göre kodlanarak sayılara çevrilir.
Hareketli görüntü	Resim bilgisine ek olarak resmin hangi resimden sonra geldiğini gösteren zaman bilgisini de içerir. Bu ek bilgi haricinde yapılan işlem resim ile aynıdır.
Ses	Ses, genlik ve frekansın zaman içinde değişimiyle kodlanır.

10 satır ve 6 sütundan oluşan “2” resmi 1 ve 0’lardan oluşan 60 uzunluğunda bir vektör ile gösterilir.

İçin 2'ye bölünür.



0	0	1	1	1	0
1	1	1	0	1	0
1	0	0	0	1	0
0	0	0	0	1	0
0	0	0	1	1	0
0	0	0	1	0	0
0	0	1	1	0	0
0	1	1	0	0	0
1	1	0	0	0	0
1	1	1	1	1	1

0 0 1 1 1 0 1 1 1 0 1 0 0 0 1 0 0 0 0 0 1 0 0 0 0 1 1 0 0 0 0 1 0 0 0 0 1 1 0 0 0 1 1 0 0 0 0 1 1 1 1 1

- Bu şekilde tanınmak istenen harf için çeşitli fontlarla yazılmış birçok örneği temsil eden 60 boyutlu vektörler elde edilir.
- Bu uygulamamız için özellik sayımız 60'tır. Diğer bir deyişle örneklerimiz 60 boyutlu bir uzayda temsil edilmektedir.
- Elimizde 10 rakama ait farklı fontlarla yazılmış 10'ar resim olursa veri kümemiz $100 \text{ örnek} * 60$ boyutlu bir matris olacaktır.
- Elimizde her örneğin hangi harf olduğunu gösteren sınıf bilgiside bulunmaktadır.
- Bu matris eğitim ve test kümesi oluşturmak için 2'ye bölünür.
- Eğitim kümesi bir sınıflandırıcıya verilir.
- Sistem modellenir.
- Modelin başarısını ölçmek için sınıflandırıcının daha önce görmediği, modelini oluşturmakta kullanmadığı test kümesi için tahminde bulunması istenir.
- Bu tahminlerle gerçek sınıfların aynılığının ölçüsü sınıflandırıcının başarı ölçüsüdür.

AKİŞ

- Makine Öğrenmesi Nedir ?
- Günlük Hayatımızdaki Uygulamaları
- Verilerin Sayısallaştırılması
- **Özellik Belirleme**
 - Özellik Seçim Metotları
 - Bilgi Kazancı (Information Gain-IG)
 - Sinyal-Gürültüye Oranı: (S2N ratio)
 - Alt küme seçiciler (Wrappers)
 - Yeni Özelliklerin Çıkarımı
 - Temel Bileşen Analizi (Principal Component Analysis)
 - Doğrusal Ayırteden Analizi (Linear Discriminant Analysis)
- Sınıflandırma Metotları
 - Doğrusal Regresyon
 - SVM (Support Vector Machine)
 - Karar Ağaçları (Decision Trees)
 - Yapay Sinir Ağları
 - En Yakın K Komşu Algoritması (k - Nearest Neighbor)
 - Öğrenmeli Vektör Kuantalama (Learning Vector Quantization)
- Kümeleme Algoritmaları
 - K-Ortalama (K-Means)
 - Kendi Kendini Düzenleyen Haritalar (Self Organizing Map -SOM)

Özellik Belirleme

- Bir doktor
- Veri: Kişi bilgilerini içeren dosyalar
- Görev: Kimlerin hasta olduğunun bulunması.
- Hangi bilgilere bakılır ?
 - Ad soyad
 - Doğum yeri
 - Cinsiyet
 - Kan tahlili sonuçları
 - Röntgen sonuçları
 - vs.

1. Özellik	2. Özellik	Sınıf
1	3	A
2	3	B
1	4	A
2	3	B

Özellik Seçimi ve Çıkarımı

Elimizdeki özellik sayısı az iken hangi özelliklerin daha doğru sınıflandırma yapacağına rahatlıkla karar verebiliriz. Özellik sayısı çok fazla iken bizim bunu gözle yapmamız imkansızdır. Bu özelliklerden bazıları sınıflandırma işleminde ayırt edici özellikler olmayabilirler. Bu durumda işin içine bilgisayarlar girmektedir.

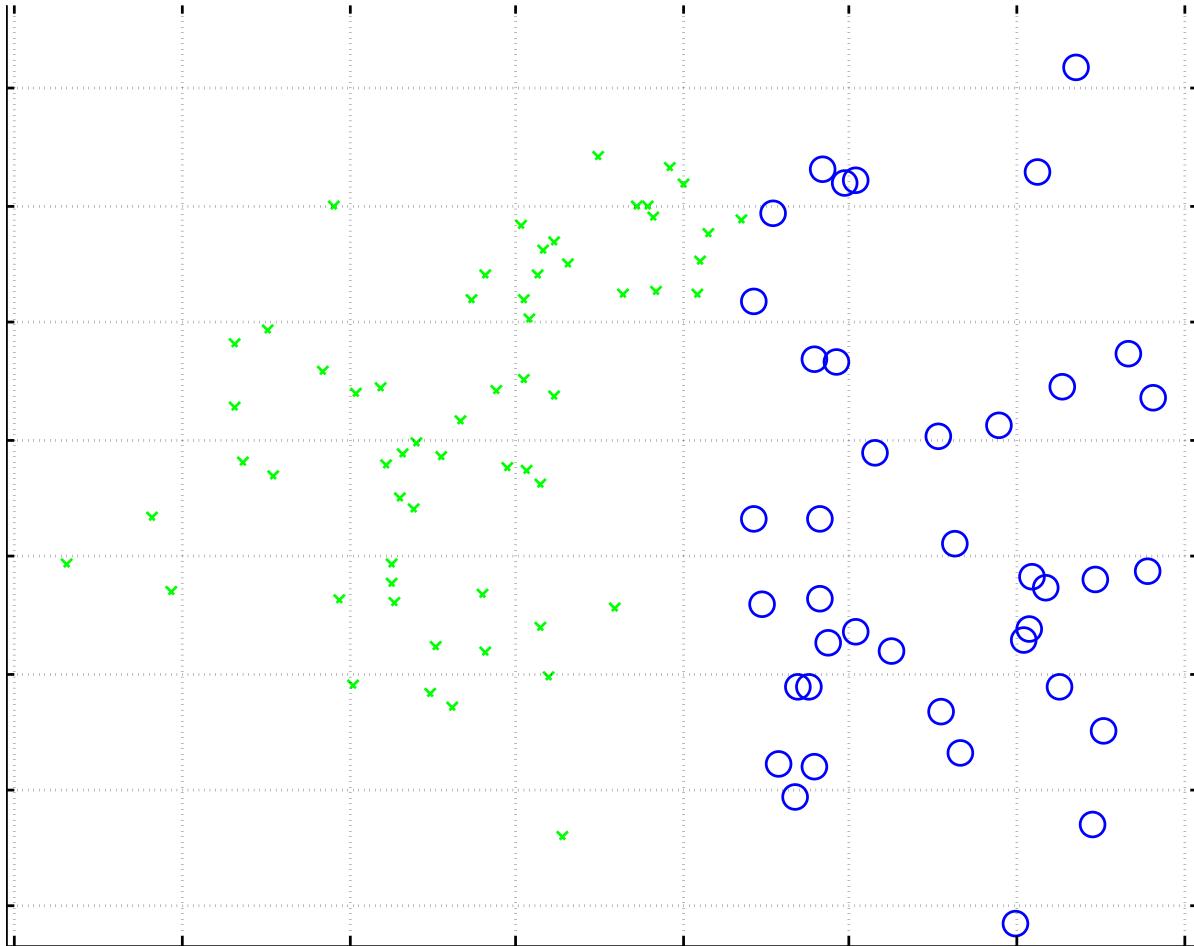
Problemi iki şekilde çözebiliriz

- Var olan özelliklerden bazılarını seçmek (özellik seçimi-feature selection)
- Var olan özelliklerin birleşiminden yeni özelliklerin çıkarılması (özellik çıkarımı-feature inference)

Özellik Seçimi

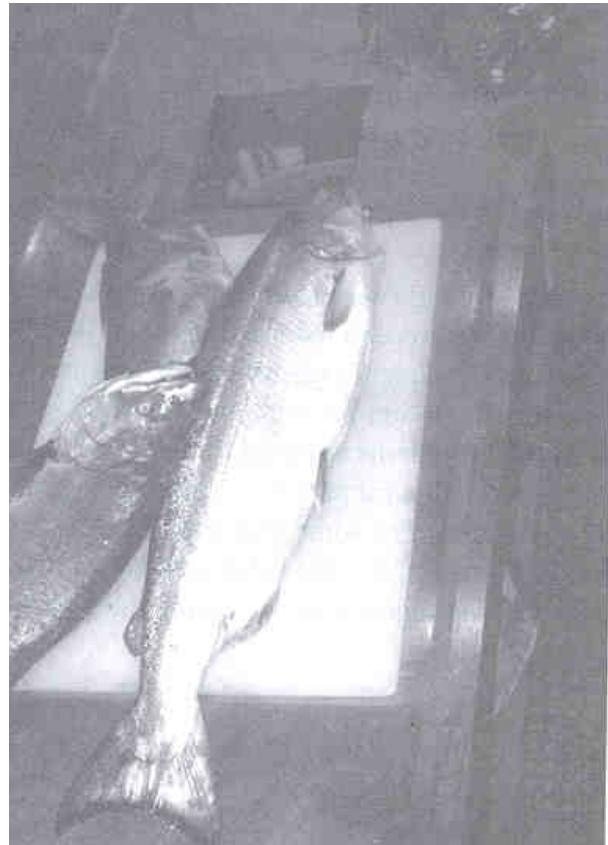
- Eğitim setindeki her bir özellik ayrı ayrı değerlendirilir.
- Seçilen özelliğin sonucu nasıl değiştirdiği incelenir.
- Etkisine göre özelliğin kullanılıp kullanılmayacağına karar verilir.

Hangi boyut ?

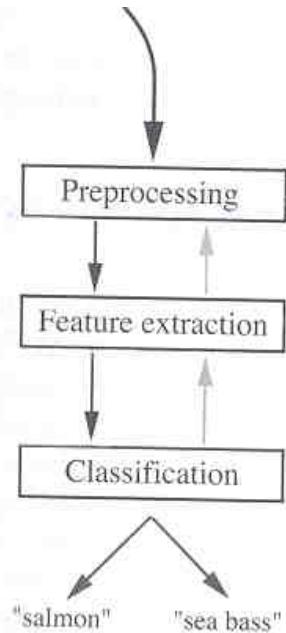


Balık Hali

- Kayan bant üzerindeki balığın türünü belirlemek (Salmon ? Sea Bass ?)

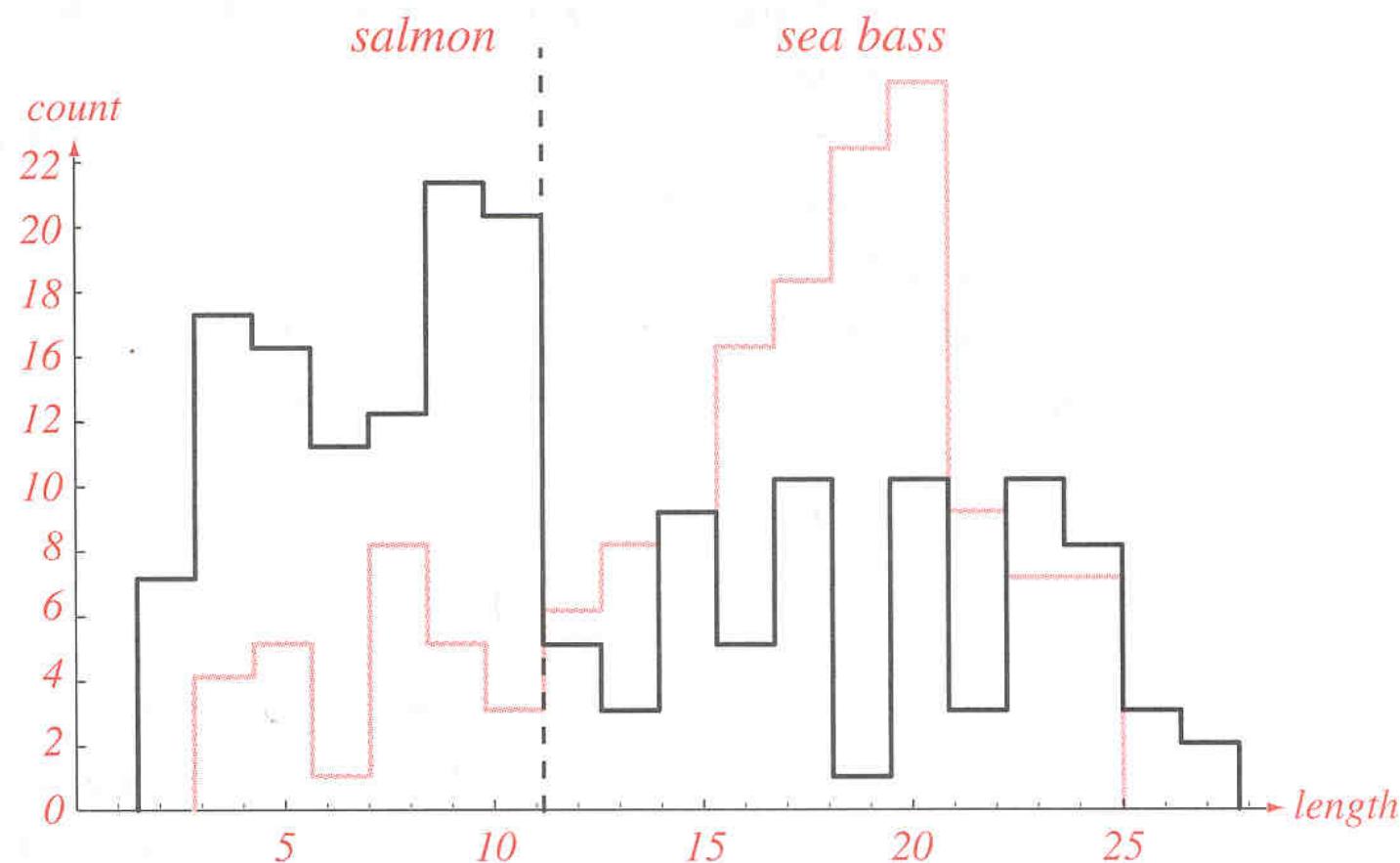


kamera



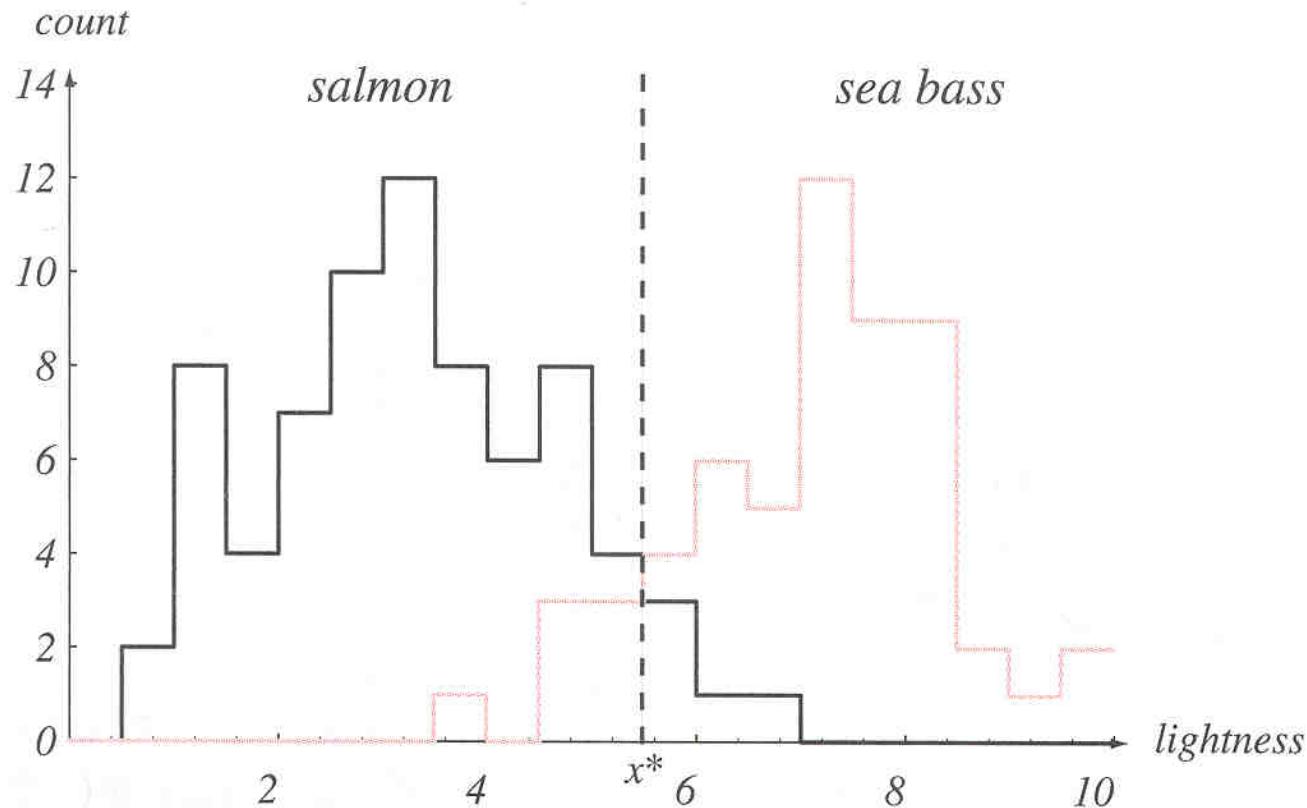
Balık Özellikleri: Uzunluk

- Salmon'lar genelde Sea Bass' lardan daha kısadırlar.

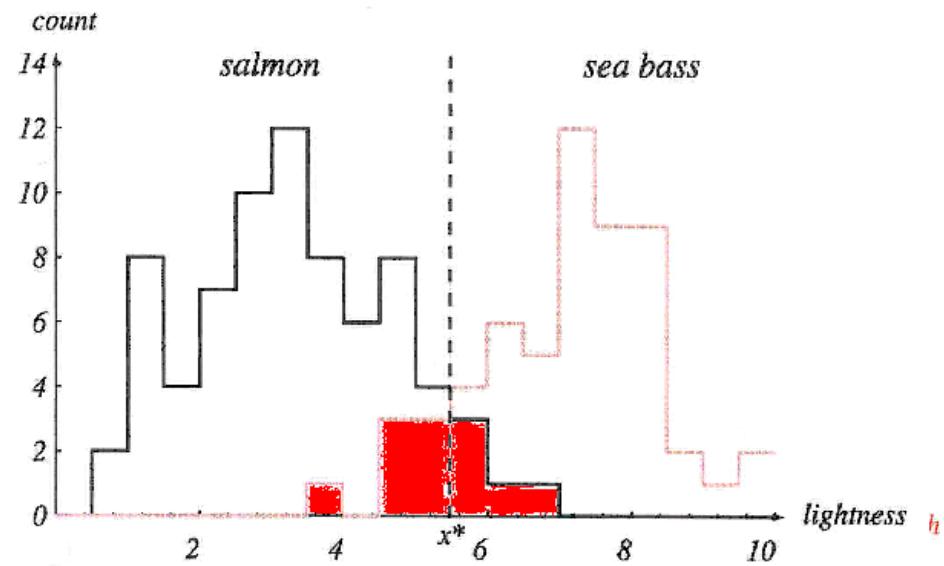
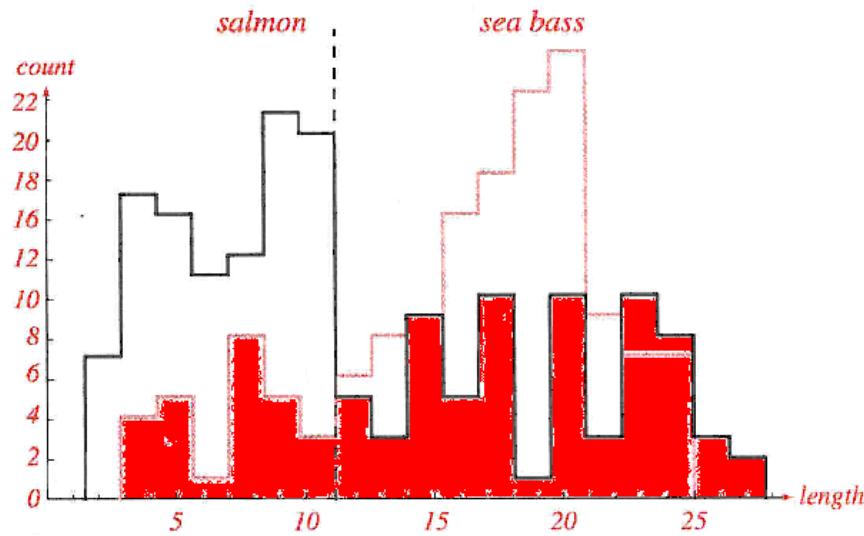


Balık Özellikleri: Parlaklık

- Sea Bass genelde Salmon'lardan daha parlaktır.



Hangi Özellik ?



Kırmızı bölgeler yapılan hataları göstermektedir.

Doktoru yoralım 😊

- Hastalık dosyasında 5000 adet özellik olsaydı ? Örneğin kişinin DNA dizisine bakarak hasta olup olmadığına karar verecek olsaydık ne yapardık ? Nerelere bakacağımıza nasıl karar verirdik ?
- Burada devreye bilgisayarları sokmamız gerekmektedir.
- Bu olay bir insanın hesap yapma kabiliyetiyle, bir hesap makinesinininkini karşılaştırmaya benziyor.

Özellik seçimi

- Bu problem makinelerle iki farklı metotla çözülebilir.
 - Var olan özelliklerden bazlarını seçmek
 - Özellikleri tek tek değerlendirmek (Filter)
 - Özellik alt kümeleri oluşturup, sınıflandırıcılar kullanıp performanslarını ölçüp, bu alt kümeleri en iyilemek için değiştirerek (Wrapper)
 - Var olan özelliklerin lineer birleşimlerinden yeni özelliklerin çıkarımı

Özellikleri birer birer inceleme (Filters)

- Eğitim verisindeki her bir özellik teker teker ele alınır.
- Örnek ile ilgili sadece o özellik elimizde olsaydı ne olurdu sorusunun cevabı bulunmaya çalışılır.
- Seçilen özellikle sınıf ya da sonucun birlikte değişimleri incelenir.
- Özellik değiştiğinde sınıf ya da sonuç ne kadar değişiyorsa, o özelliğin sonuca o kadar etkisi vardır denilir.

I - Bilgi Teorisi (Information Gain)

En iyi sınıflandırmayı yapan özellik nasıl seçilir ?

I- Information Gain

Her bir özelliğin Bilgi Kazancı (*information gain*) hesaplanır.
Negatif ve pozitif örneklerden oluşan bir S kümesi olsun.
 S kümescinin Entropy'si hesaplanırken

$$\text{Entropy}(S) = - p \log p - q \log q \quad \text{kullanılır.}$$

S kümescinde 14 örnek olsun: 9 pozitif ve 5 negatif

$$\text{Entropy}(S) = - (9/14) \log (9/14) - (5/14) \log (5/14) = 0.94$$



Daha önceki hava, nem, rüzgar, su sıcaklığı gibi değerlere göre pikniğe gidip gitmemeye kararı verilmiş 4 olay

Olay No	Hava	Nem	Rüzgar	Su sıcaklığı	Pikniğe gidildi mi?
1	güneşli	normal	güçlü	ılık	Evet
2	güneşli	yüksek	güçlü	ılık	Evet
3	yağmurlu	yüksek	güçlü	ılık	Hayır
4	güneşli	yüksek	güçlü	soğuk	Evet

$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

Olay No	Hava	Nem	Rüzgar	Su sıcaklığı	Pikniğe gidildi mi?
1	güneşli	normal	güçlü	ılık	Evet
2	güneşli	yüksek	güçlü	ılık	Evet
3	yağmurlu	yüksek	güçlü	ılık	Hayır
4	güneşli	yüksek	güçlü	soğuk	Evet

Pikniğe gidildi mi? sorusunun iki cevabı vardır.

Evet cevabının olasılığı $\frac{3}{4}$

Hayır cevabının olasılığı $\frac{1}{4}$

$$E(\text{Piknik}) = -(3/4) \log_2(3/4) - (1/4) \log_2(1/4) = 0.811$$

Her özelliğin, her değeri için bilgi kazancı hesaplanır.

$$\text{Gain(Piknik,Hava)} = 0.811 - \left(\frac{3}{4} \right) \left(-\frac{3}{3} \log_2 \left(\frac{3}{3} \right) - 0 \right) - \left(\frac{1}{4} \right) \left(0 - \left(\frac{1}{1} \log_2 \left(\frac{1}{1} \right) \right) \right) = 0.811$$

Aşağıda Hava özelliğinin IG'si hesaplanırken bulunan rakamların açıklamaları verilmiştir.

0.811 → Pikniğe gitme olayının Entropisi

(3/4) → havanın güneşli olma oranı

(3/3) → hava güneşli iken pikniğe gidilme oranı

0 → hava güneşli iken pikniğe gidilmeme oranı

(1/4) → havanın yağmurlu olma oranı

0 → hava yağmurlu iken pikniğe gidilme oranı

(1/1) → hava yağmurlu iken pikniğe gidilmeme oranı

$$\begin{aligned}\text{Gain(Piknik,Nem)} &= 0.811 - \left(\frac{1}{4} \right) \left(-\frac{1}{1} \log_2 \left(\frac{1}{1} \right) - 0 \right) - \left(\frac{3}{4} \right) \left(-\frac{2}{3} \log_2 \left(\frac{2}{3} \right) - \left(\frac{1}{3} \log_2 \left(\frac{1}{3} \right) \right) \right) \\ &= 0.811 - 0.688 = 0.1225\end{aligned}$$

$$\begin{aligned}\text{Gain(Piknik,Rüzgar)} &= 0.811 - \left(\frac{4}{4} \right) \left(-\frac{3}{4} \log_2 \left(\frac{3}{4} \right) - \left(\frac{1}{4} \log_2 \left(\frac{1}{4} \right) \right) \right) \\ &= 0.811 - 0.811 = 0\end{aligned}$$

$$\begin{aligned}\text{Gain(Piknik,SuSıcaklığı)} &= 0.811 - \left(\frac{3}{4} \right) \left(-\frac{2}{3} \log_2 \left(\frac{2}{3} \right) - \left(\frac{1}{3} \log_2 \left(\frac{1}{3} \right) \right) \right) - \left(\frac{1}{4} \right) \left(-\frac{1}{1} \log_2 \left(\frac{1}{1} \right) \right) \\ &= 0.811 - 0.688 = 0.1225\end{aligned}$$

Bilgi Kazancı en büyük olan özellik **hava dir. Gerçek uygulamalarda ise yüzlerce özelliğin IG hesaplanır ve en büyük olanları seçilir.**

• II- Sinyalin gürültüye oranı (S2N ratio)

Sınıflar arası ayrılıkların fazla, sınıf içi ayrılıkların az olduğu özellikler seçilir.

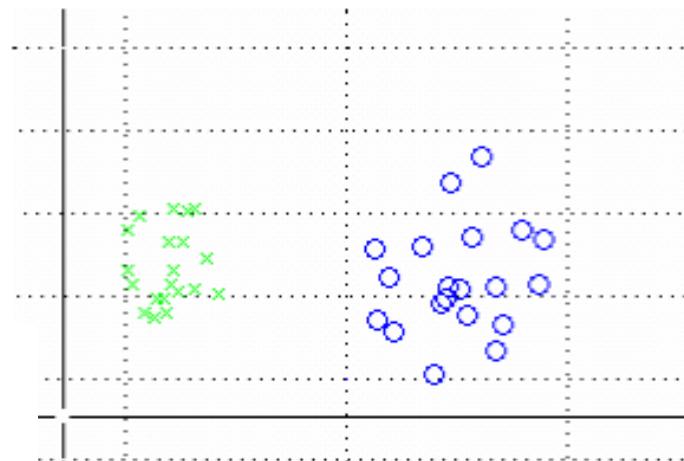
$$S_i = \frac{m_1 - m_2}{d_1 - d_2}$$

m_1 → sınıf1'deki i. özelliklerin ortalaması

m_2 → sınıf2'deki i. özelliklerin ortalaması

d_1 → sınıf1'deki i. özelliklerin standart sapması

d_2 → sınıf2'deki i. özelliklerin standart sapması

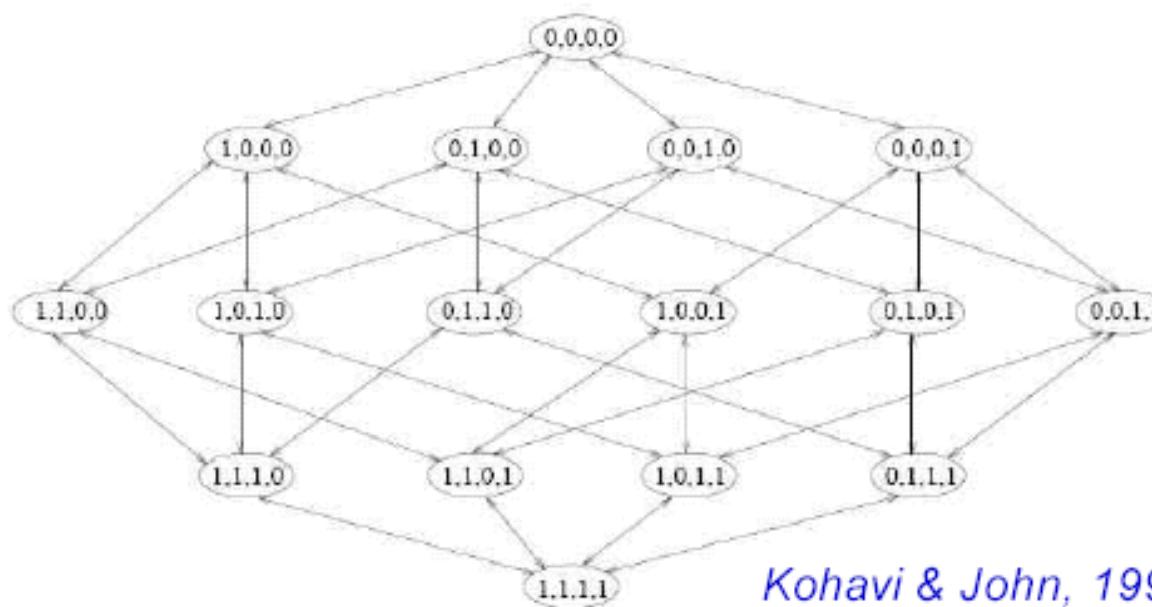


- S2N oranı her bir özellik için ayrı ayrı hesaplanır.
- S değeri en yüksek olan özellikler seçilerek kullanılır.

• III- Duyarlılık Analizi

Özellik altkümesi seçiciler (Wrappers)

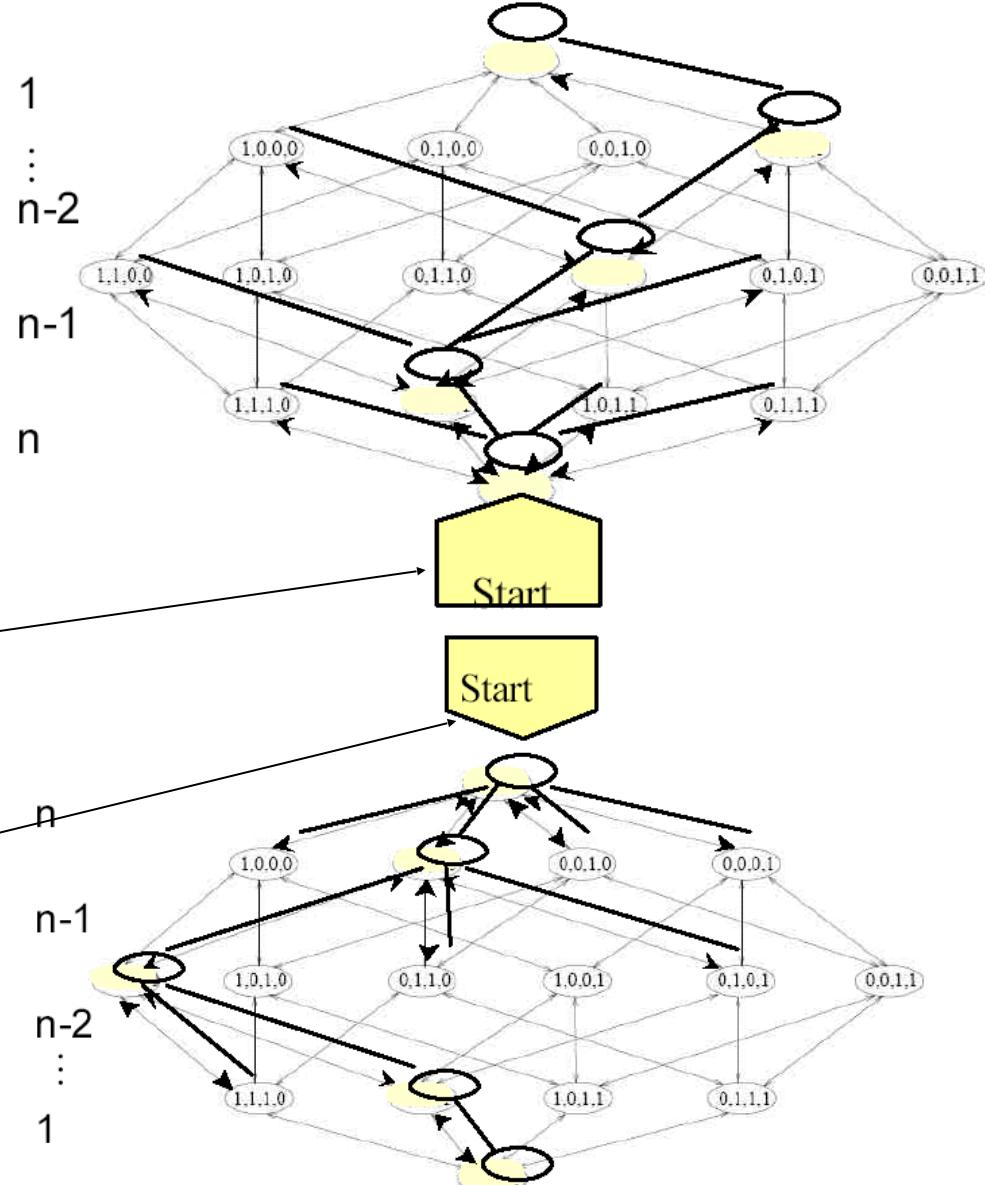
Her bir özellik için test yapılırken, test edilen özellik dışındaki bütün özellikler sabit tutularak test edilen özelliğin değerindeki değişimlerin sınıflandırma/kümeleme/regresyon sonuçlarına göre etkisi ölçülür. En çok etki yapan özellik seçilir.



N özellik için olası 2^N özellik alt kümesi = 2^N eğitim

Özellik altkümesi seçiciler

- *Hızlandırmak için tüm olasılıkları denemek yerine*
 - Hepsiyle başlayıp her seferinde bir tane elemek
 - Tek özellikle başlayıp her seferinde bir tane eklemek



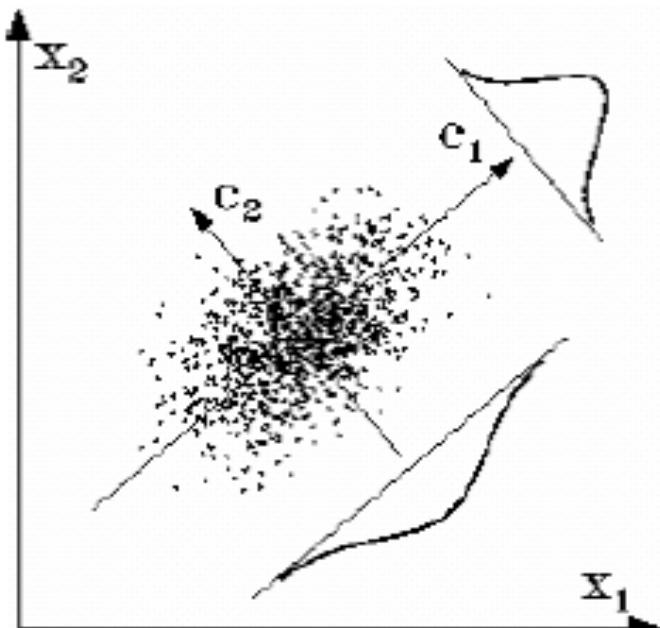
Hangi yoldan gidileceğine o özellik kümelerinin sınıflandırmadaki performansına bakılarak karar verilir.

Yeni Özelliklerin Çıkarımı

- Var olan özelliklerin lineer birleşimlerinden yeni bir özellik uzayı oluşturular ve veriler bu uzayda ifade edilirler. Yaygın olarak kullanılan 2 metot vardır.
- PCA
- LDA

Temel Bileşen Analizi (Principle Component Analysis-PCA)

- Örneklerin en fazla değişim gösterdiği boyutlar bulunur.
- Soldaki şekilde veriler c_1 ve c_2 eksenlerine izdüşümleri alındığındaki dağılımları gösterilmiştir.
- C_1 eksenindeki değişim daha büyütür.
- Böylece veriler 2 boyuttan tek bir boyuta c_1 eksenine indirgenmiş olur.

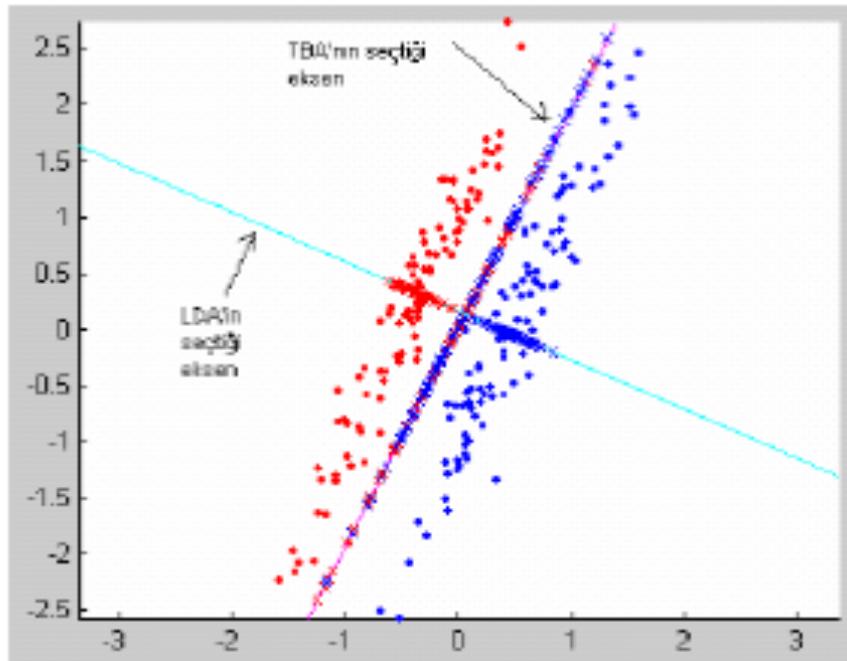


Temel Bileşen Analizi'nın Adımları

- N boyutlu verinin NxN boyutlu Kovaryans matrisi bulunur
- Matrisin N adet özdeğeri (eigen value) bulunur
- En büyük ilk M özdeğere karşılık gelen M adet öz vektör (eigenvector) bulunur.
- Veriler M öz vektöre izdüşürülerek N boyuttan M boyuta indirgenmiş olur.

Öz değerler, veriler o özdeğere karşılık gelen özvektöre izdüşüm yapıldığındaki verinin varyansıdır. En büyük varyansa sahip olmak en fazla değişimi göstermek olduğundan öz değerlerin en büyükleri seçilerek işlem gerçekleştirilir.

Doğrusal Ayırteden Analizi (Linear Discriminant Analysis–LDA)



- PCA, verilerin sınıflarına bakmadan boyut indirgeme işlemini gerçekleştirir.
- Soldaki şekilde görüldüğü gibi bazı durumlarda sınıf örnekleri birbirinin içerisine girdiği için sınıflandırma başarısı düşer.
- Bu gibi durumlarda LDA kullanılır. LDA varyans değerlerine ek olarak sınıf bilgisini de kullanarak boyut indirgeme yapar.

Sınıflandırma Yöntemleri

Çok fazla sayıda sınıflandırma yöntemi mevcuttur.

Niye bu kadar çok metot var?
Ne zaman hangisini kullanacağız?

Her veri kümesi üzerinde mükemmel çalışan bir yöntem olmadığından buna ihtiyaç vardır.

Literatürde en fazla sıklıkla kullanılan yöntemler

Destek Vektör Makineleri (SVM-Support Vector Machine)

Yapay Sinir Ağları (Artificial Neural Network)

Karar Ağaçları (Decision Tree)

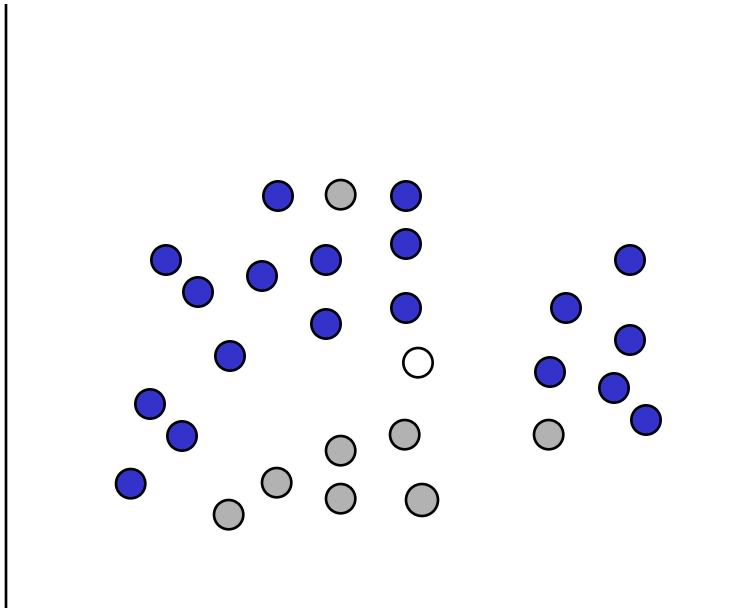
K-En Yakın Komşuluk (K- Nearest Neighbor / KNN)

AKİŞ

- Makine Öğrenmesi Nedir ?
- Günlük Hayatımızdaki Uygulamaları
- Verilerin Sayısallaştırılması
- Özellik Belirleme
 - Özellik Seçim Metotları
 - Bilgi Kazancı (Information Gain-IG)
 - Sinyal-Gürültüye Oranı: (S/N ratio)
 - Alt küme seçiciler (Wrappers)
 - Yeni Özelliklerin Çıkarımı
 - Temel Bileşen Analizi (Principal Component Analysis)
 - Doğrusal Ayırteden Analizi (Linear Discriminant Analysis)
- **Sınıflandırma Metotları**
 - Doğrusal Regresyon
 - SVM (Support Vector Machine)
 - Karar Ağaçları (Decision Trees)
 - Yapay Sinir Ağları
 - En Yakın K Komşu Algoritması (k - Nearest Neighbor)
 - Öğrenmeli Vektör Kuantalama (Learning Vector Quantization)
- **Kümeleme Algoritmaları**
 - K-Ortalama (K-Means)
 - Kendi Kendini Düzenleyen Haritalar (Self Organizing Map -SOM)

Sınıflandırma Metotları

Görev: Önceden etiketlenmiş örnekleri kullanarak yeni örneklerin sınıflarını bulmak

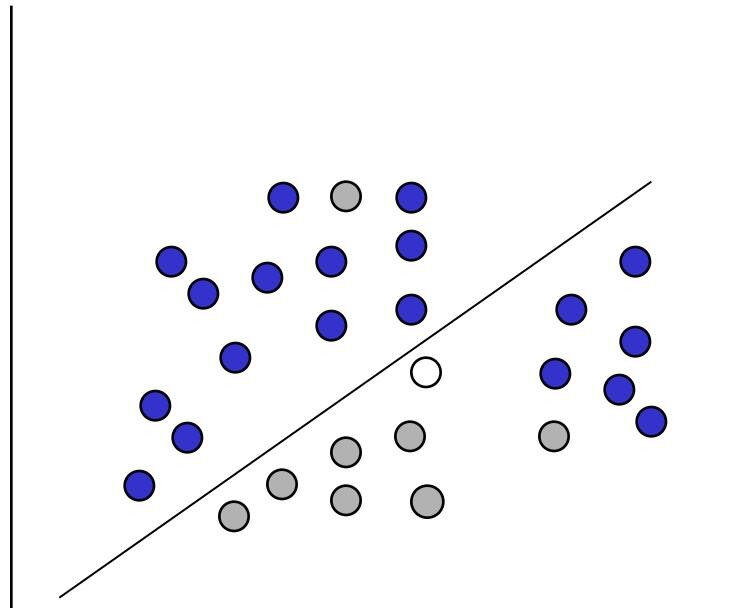


Mavi ve gri sınıfından örnekler
Beyaz, mavi mi gri mi?



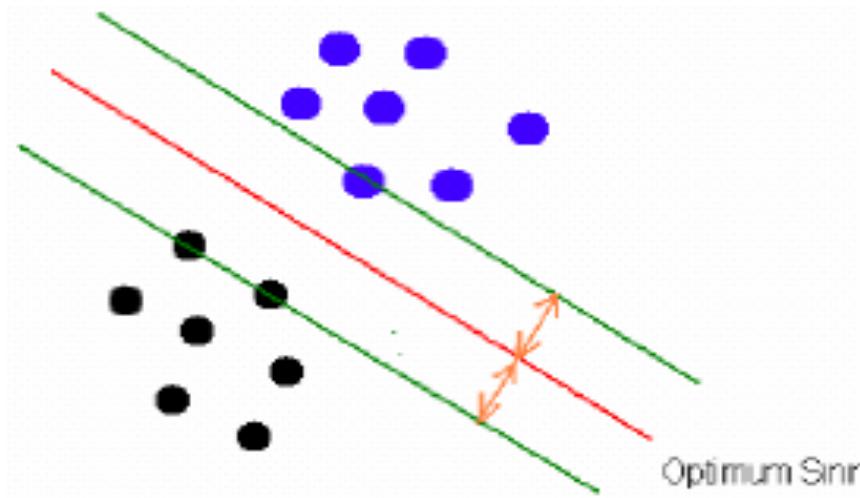
Metotlar:
Regresyon,
SVM,
Karar Ağaçları,
LVQ,
Yapay Sinir Ağları,
...

Doğrusal Regresyon

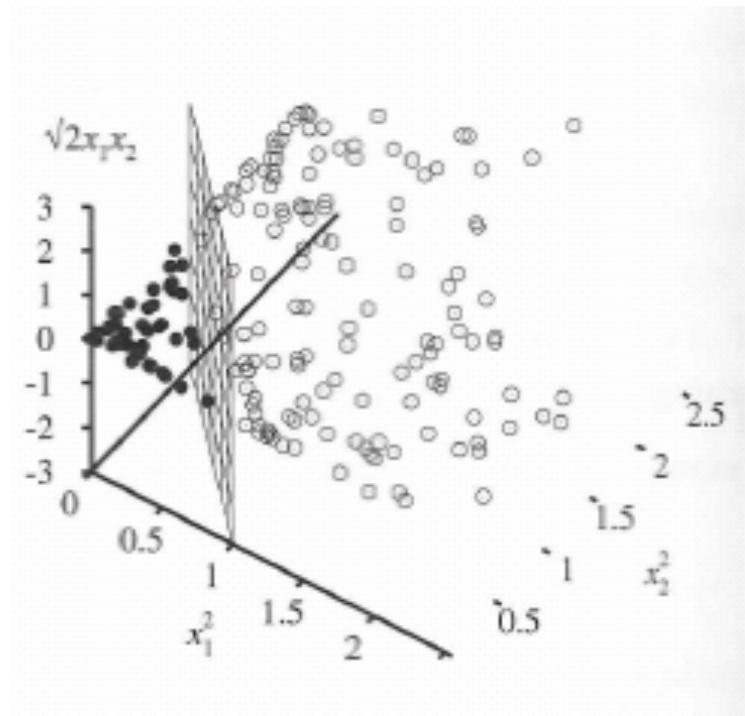
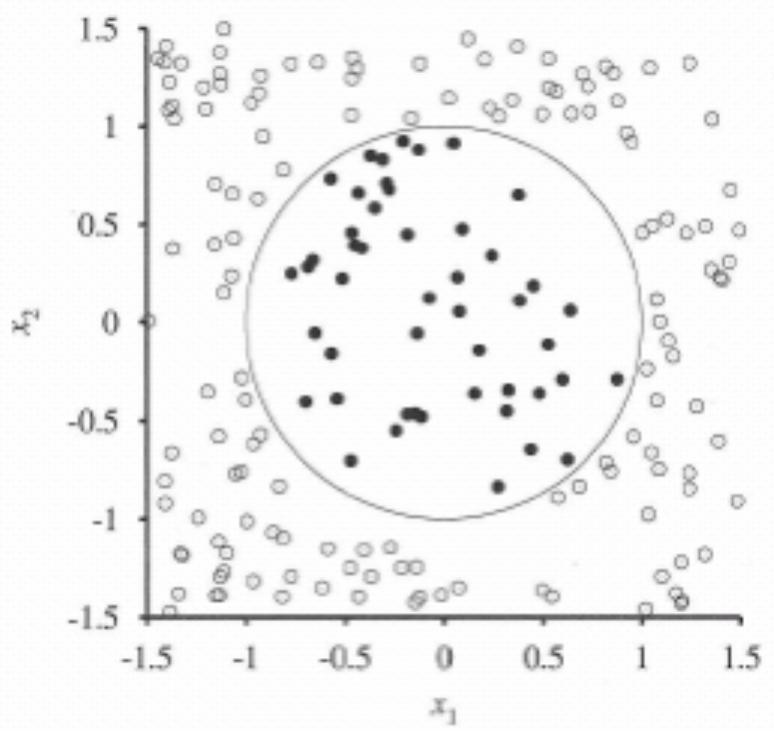


- $w_0 + w_1 x + w_2 y \geq 0$
- Regresyon en az hata yapan w_i leri bulmaya çalışır.
- Basit bir model
- Yeterince esnek değil

Destek Vektör Makineleri (SVM-Support Vector Machine)

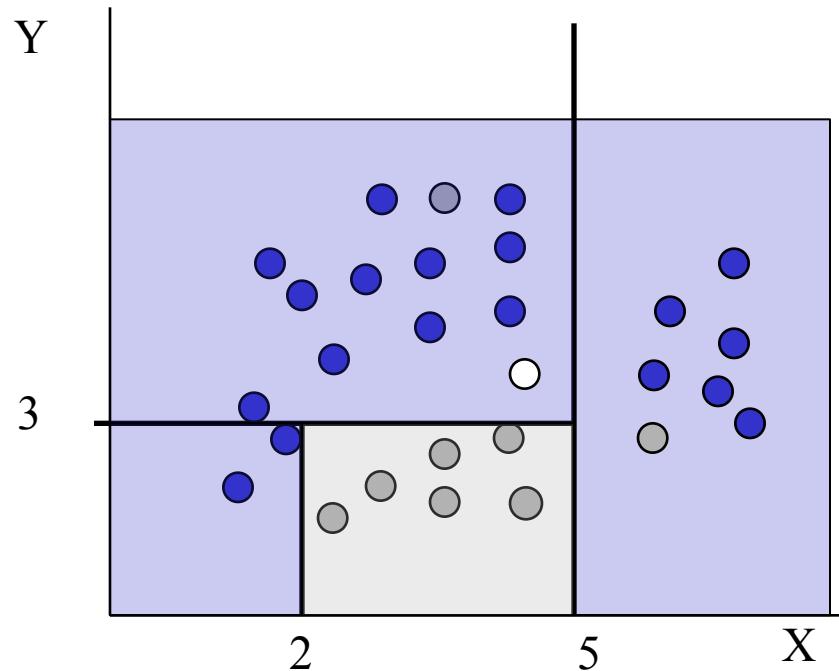


- Sınıfları birbirinden ayıran özel bir çizginin (hyperplane)ının bulunmasını amaçlar. SVM, her iki sınıfı da en uzak olan hyperplane bulmayı amaçlar.
- Eğitim verileri kullanılarak hyperplane bulunduktan sonra, test verileri sınırın hangi tarafında kalmışsa o sınıfı dahil edilir.



- Lineer olarak ayrılamayan örneklerde veriler daha yüksek boyutlu başka bir uzaya taşınır ve sınıflandırma o uzayda yapılır.
- Soldaki şekilde örnekler lineer olarak ayrılamaz iken, sağdaki şekilde üç boyutlu uzayda (x_1^2 , x_2^2 , $\sqrt{2}x_1x_2$) ayrılabilmektedir.

Karar Ağaçları



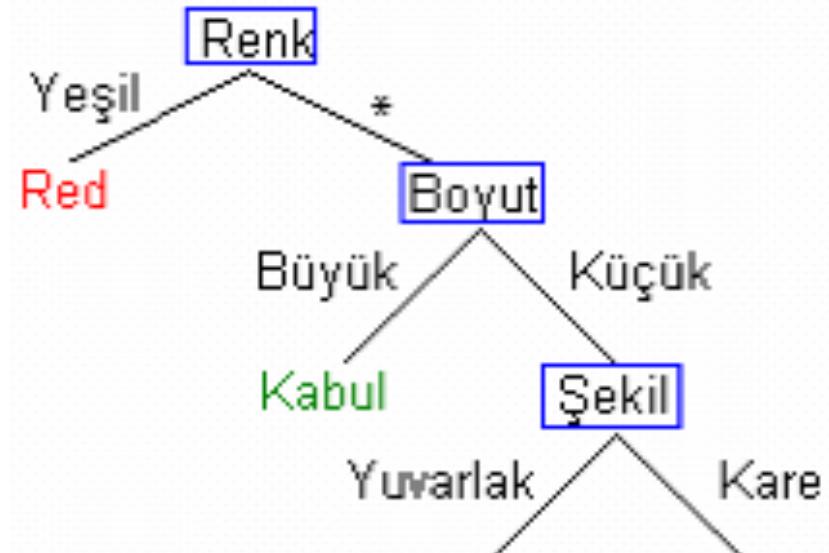
Böl ve yönet stratejisi

Nasıl böleceğiz ?

```
if X > 5 then blue  
else if Y > 3 then blue  
else if X > 2 then grey  
else blue
```

Karar Ağaçları

- Ürettikleri kurallar anlaşılır.
- Karar düğümleri ve yapraklardan oluşan hiyerarşik bir yapı.



<u>Şekil</u>	<u>Renk</u>	<u>Boyut</u>	<u>Sınıf</u>
Yuvarlak	Yeşil	Küçük	Red
Kare	Siyah	Büyük	Kabul
Kare	Sarı	Büyük	Kabul
Yuvarlak	Sarı	Küçük	Red
Kare	Yeşil	Büyük	Red
Kare	Sarı	Küçük	Kabul

Karar Ağaçları Oluşturma

- Tüm veri kümesiyle başlanır.
- Bir özelliğin bir değerine göre veri kümesi iki alt kümeye bölünür. Bölmede kullanılan özellikler ve değerler karar düğüme yerleştirilir.
- Her alt küme için aynı prosedür, her alt kümede sadece tek bir sınıfa ait örnekler kalıncaya kadar uygulanır.



Karar Düğümleri Nasıl Bulunur ?

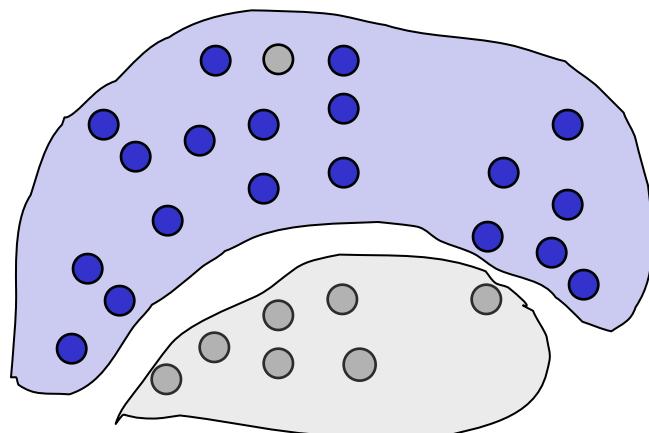
- Karar düğümlerinde yer alan özelliğin ve eşik değerinin belirlenmesinde genel olarak **entropi** kavramı kullanılır.
- Eğitim verisi her bir özelliğin her bir değeri için ikiye bölünür. Oluşan iki alt kümenin entropileri toplanır. En düşük entropi toplamına sahip olan özellik ve değeri karar düğümüne yerleştirilir.

Karar Ağaçlarıyla Sınıflandırma

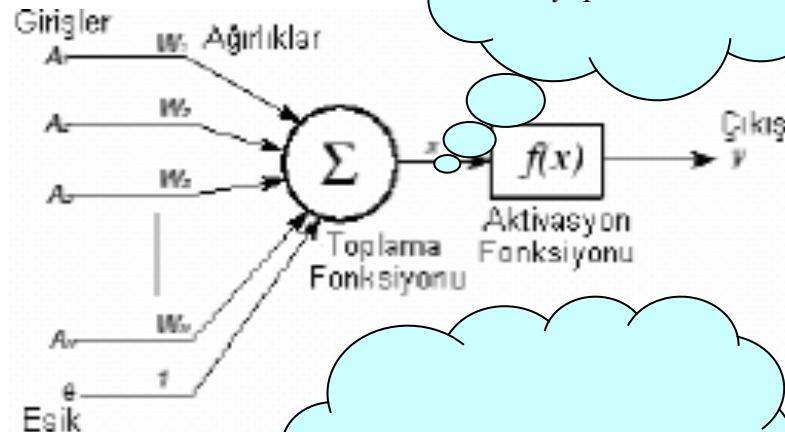
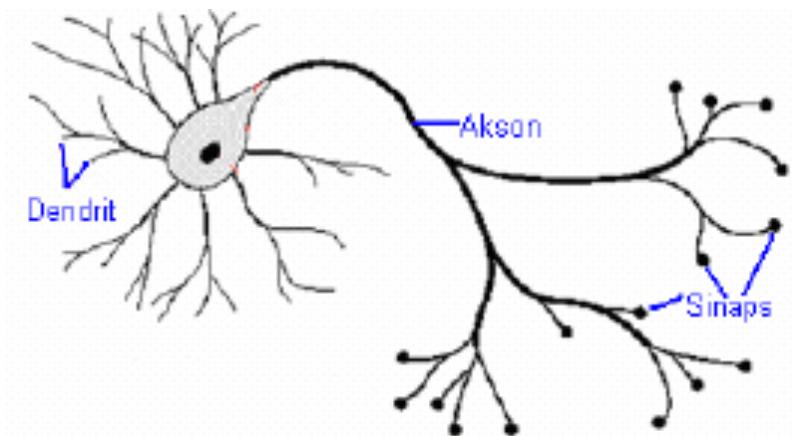
- En tepedeki kök karar düğümünden başla.
- Bir yaprağa gelinceye kadar karar düğümlerindeki yönlendirmelere göre dallarda ilerle (Karar düğümlerinde tek bir özelliğin adı ve bir eşik değeri yer alır. O düşüme gelen verinin hangi dala gideceğine verinin o düğümdeki özelliğinin eşik değerinden büyük ya da küçük olmasına göre karar verilir).
- Verinin sınıfı, yaprağın temsil ettiği sınıf olarak belirle.

Yapay Sinir Ağları

Canlılardaki sinir hücreleri ve ağları modellenerek yapay sinir ağları oluşturulmuştur.



Yapay Sinir Ağları

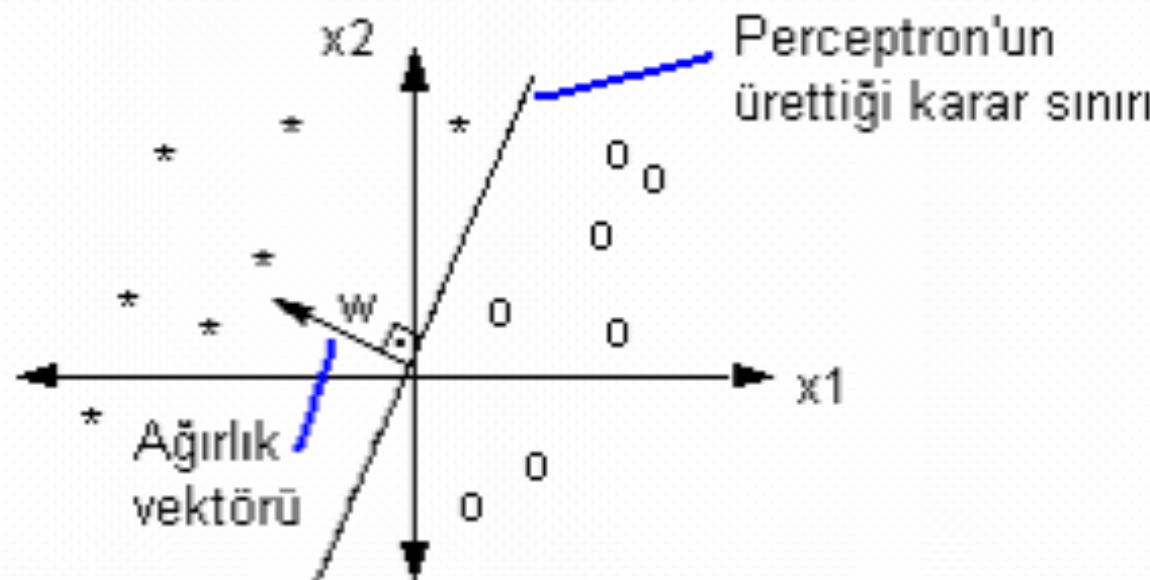


$$x = \sum_{i=1}^n A_i x W_i + \theta$$

$$y = \frac{1}{1 + \exp(-x)}$$

Gerçek sinir hücreleri, dentritlerden gelen sinyaller belirli bir eşik değerinin üzerine çıktığında akson'lar yardımıyla komşu hücrelere iletilir. Yapay hücrelerde de bu modellenir. Sinyal girişleri (A_i) ve bunları toplayan bir birim giriş sinyallerinin (A_i) ağırlıkları ile (W_i) çarpımlarını toplayan ve bu toplama eşik değerini de ekleyip bir aktivasyon fonksiyonundan geçirerek çıkış elde eder.

Tek yapay sinir hücresına **perceptron** denir.



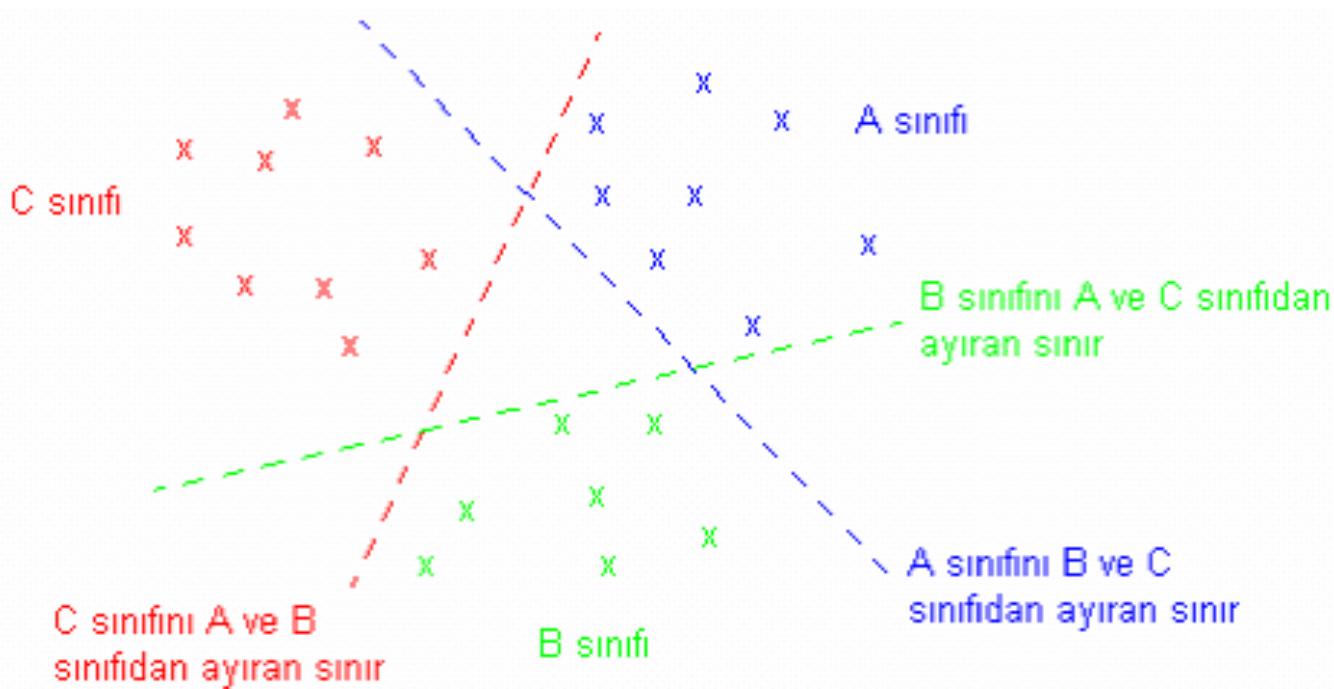
Perceptronların ağırlık değerlerinin belirlenmesi:

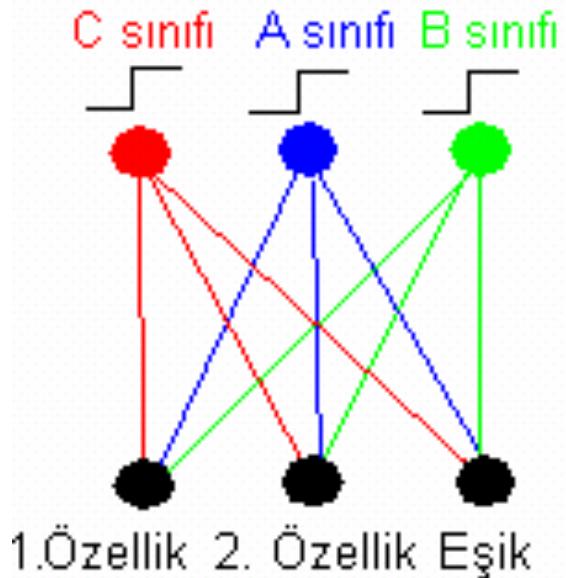
1. Ağırlıklara rasgele ilk değerler atanır.
2. (0-1) arasında öğrenme katsayıısı (μ) seçilir.
3. Ağırlıklar değiştiği sürece Her bir eğitim örneği için (x, t):

($x \rightarrow$ örneğin değerlerini; $t \rightarrow$ sınıfını, $w \rightarrow$ ağırlıkları gösterir)

- Çıkışı hesapla. $y = f(w^*x + \text{esik})$
- Çıkışla (perceptronun cevabıyla) gerçek sınıf aynı ise ($y = t$) ağırlıkları değiştirme
- Farklılarsa ($y \neq t$), ağırlıkları $w(\text{yeni}) = w(\text{eski}) + \mu (t - y) x$ güncelle.

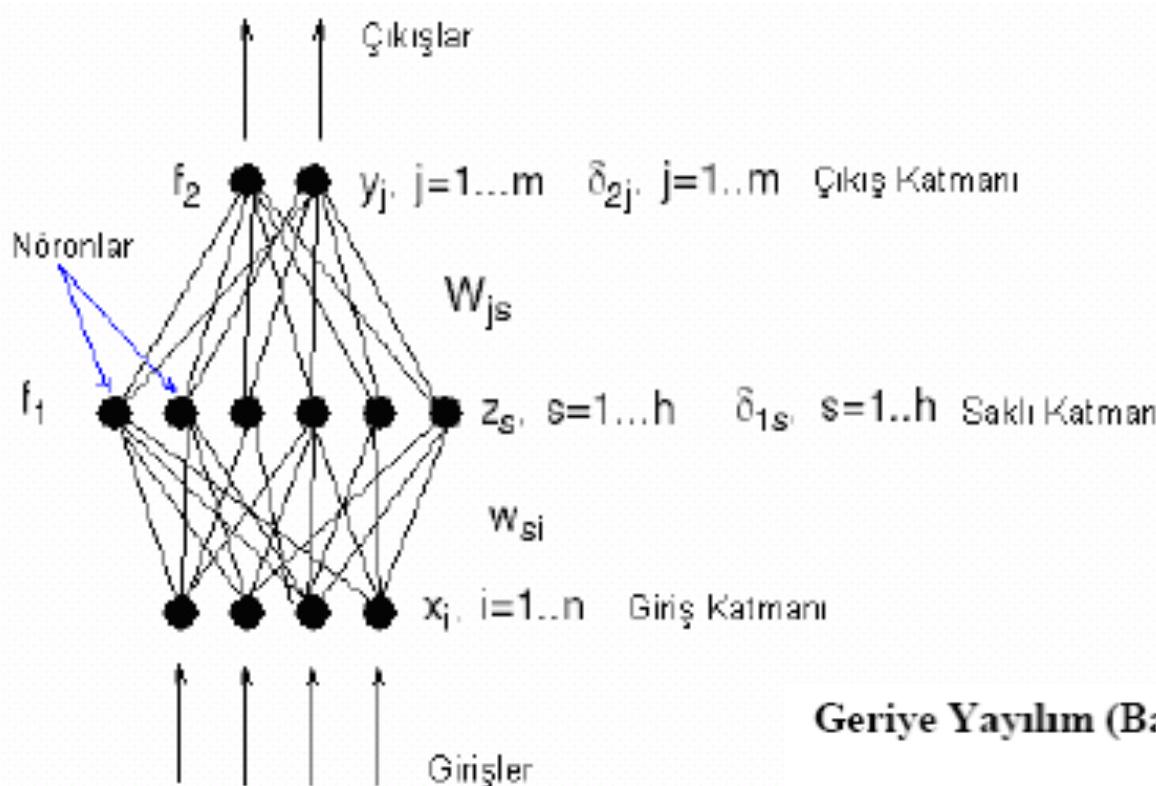
- İkiden fazla sınıfı birbirinden ayırmak için perceptron katmanı oluşturmak gereklidir. Şekilde 3 sınıfından oluşan bir veri kümesi ve bu veriyi sınıflandıran perceptron katmanı görülmektedir. Her bir sınıfı diğer sınıflardan ayırt edebilmek için perceptron kullanılmıştır.





Doğrusal olmayan karar sınırları üretebilmek için çok katmanlı perceptronlar kullanılır.

Çok katmanlı perceptronlar genelikle geriye yayılım algoritması ile eğitilirler.



Geriye Yayılım (Backpropagation) algoritması

- $x \rightarrow$ bir eğitim örneği
- $n \rightarrow$ örneklerin boyutu
- $h \rightarrow$ saklı katmandaki nöron sayısı
- $m \rightarrow$ çıkış katmanındaki nöron sayısı
- $f_1 \rightarrow$ saklı katmandaki aktivasyon fonksiyonu
- $f_2 \rightarrow$ çıkış katmanındaki aktivasyon fonksiyonu
- $z \rightarrow$ saklı katmanın çıkışları
- $y \rightarrow$ ağın çıkışları

Eğitim setindeki her bir örnek için aşağıdaki 3 adımlın tekrarlanması bir çevrim (epoch) adı verilir. Sistemin eğitimine önceden belirlenmiş bir hata değerine ulaşılıncaya kadar ya da maksimum çevrim sayısına erişilinceye kadar devam edilir.

1.Adım: İleri Yayılım

Her saklı nöron için net_i ve y_i hesaplanır, $i=1, \dots, h$:

$$\text{net}_i = \sum_{r=1}^n w_{ri} x_r \quad z_i = f_1(\text{net}_i)$$

Her çıkış nöronu için net_j ve y_j hesaplanır, $j=1, \dots, m$:

$$\text{net}_j = \sum_{i=1}^h w_{ij} z_i \quad y_j = f_2(\text{net}_j)$$

Step 2: Geri Yayılım

Her çıkış nöronu için hata hesaplanır, $j=1, \dots, m$:

$$\delta_{2j} = (t_j - y_j) f'_2(\text{net}_j)$$

Her saklı nöron için hata hesaplanır, $i=1, \dots, h$:

$$\delta_{1i} = f'_1(\text{net}_i) \sum_{j=1}^m w_{ij} \delta_{2j}$$

Step 3: Ağırlıklar güncellenir:

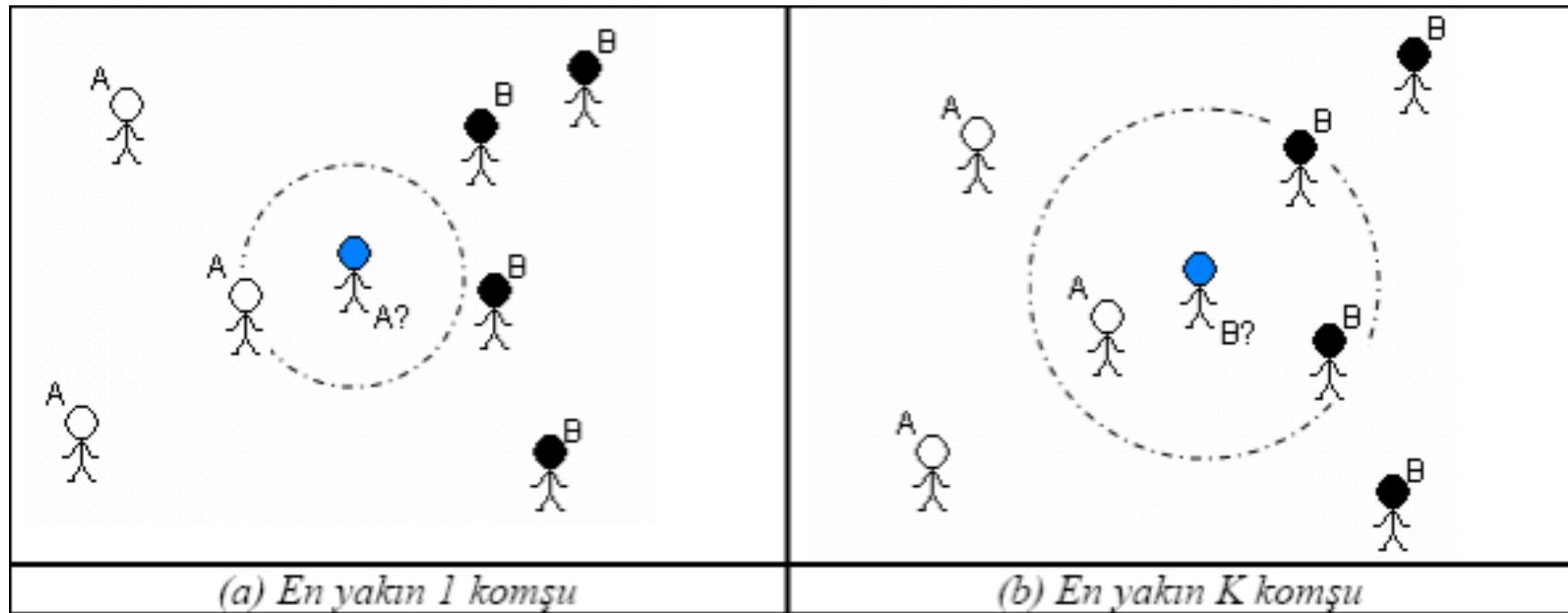
$$w_{ij}(\text{yeni}) = w_{ij}(\text{eski}) - \mu \delta_{2j} z_i$$

$$w_{ri}(\text{yeni}) = w_{ri}(\text{eski}) - \mu \delta_{1i} x_r$$

K-En Yakın Komşuluk (K- Nearest Neighbor – KNN)

K-En Yakın Komşu / K-NN algoritması, eğiticili ve örnek tabanlı (instance based) bir sınıflandırma algoritmasıdır. Bu tip algoritmalarla eğitim işlemi yapılmaz. Test edilecek örnek, eğitim kümesindeki her bir örnek ile bire bir işleme alınır.

(Bana arkadaşını söyle, sana kim olduğunu söyleyeyim)



Bir test örneğinin sınıfı belirlenirken eğitim kümesinde o örneğe en yakın K adet örnek seçilir. Seçilen örnekler içerisinde en çok örneği bulunan sınıf, test örneğinin sınıfı olarak belirlenir.

$$y(x_q) = \arg \max_{t \in C} \sum_{j=1}^k \delta(x_j, c_t)$$

Örnekler arasındaki uzaklık hesaplanırken eulidean distance kullanılır.

$$d(x_i, x_j) = \sqrt{\sum_{r=1}^n (a_r(x_i) - a_r(x_j))^2}$$

Öğrenmeli Vektör Kuantalama (Learning Vector Quantization)

[η] öğrenme oranı

[n] maximum eğitim sayısı

[c] betimleyici vector sayısı

[μ_1, \dots, μ_c] betimleyici vektörler (centroids)

[x] eğitim datasından bir örnek

[S(x)] x vektörünün ait olduğu yada betimlediği sınıf
olmak üzere

1. $\eta, n, \mu_1, \dots, \mu_c$ için ilk değer atamalarını gerçekleştirir

2. Aşağıdaki işlemleri n defa tekrar et

2.1 x eğitim datasını al

2.2 x e en yakın betimleyici vektörü bul

$$(\mu_k) : k \leftarrow \arg \min_j \|x - \mu_j\| \quad j=1 \dots c$$

2.3 μ_k nin güncellenmesi:

Eğer x doğru sınıfı ($s(x)=s(\mu_k)$) sınıfları aynı ise)

$\mu_k \leftarrow \mu_k + \eta(x - \mu_k)$ ödüllendir x' e yaklaşır

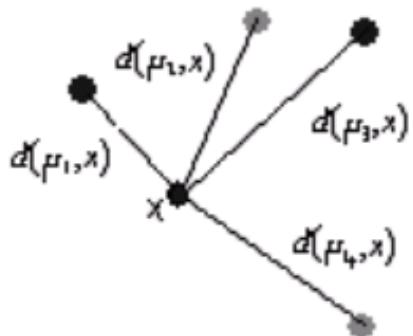
değilse

$\mu_k \leftarrow \mu_k - \eta(x - \mu_k)$ cezalandır x' den uzaklaştır

LVQ'da eğitim

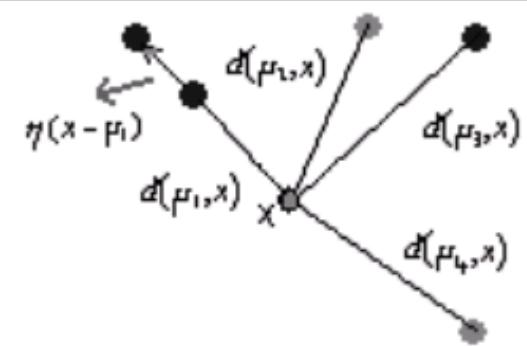
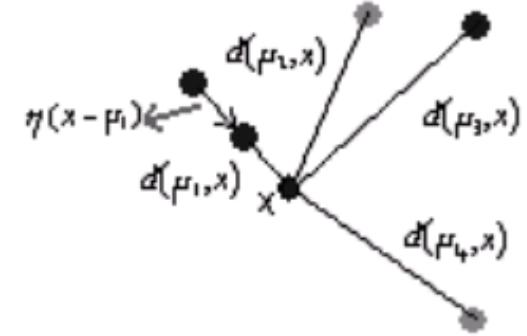
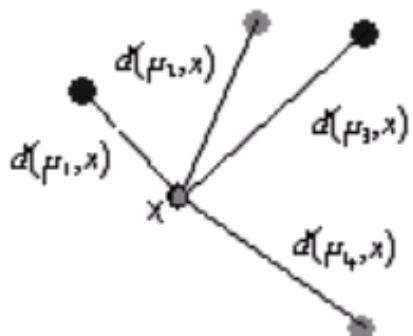
LVQ'da ödüllendirme

Kazanan vektörle, örnek
aynı sınıfından (ikisi de siyah
sınıftan)



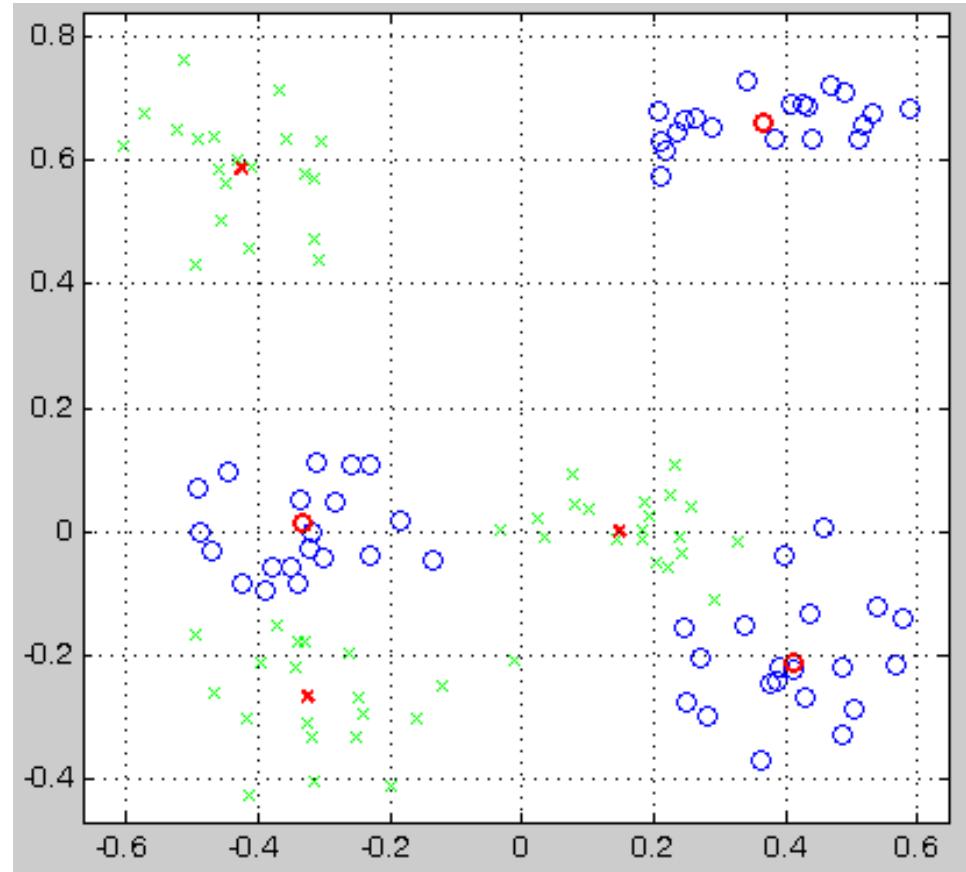
LVQ'da cezalandırma

Kazanan vektörle, örnek
farklı sınıflardan (kazanan
siyah, örnek gri sınıfından)



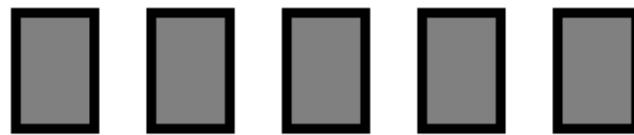
LVQ- Test İşlemi

- Eğitim sonucu bulunan 2 sınıfı ait 3'er betimleyici vektör.
- Test işlemi, test örneğinin bu 6 vektörden en yakın olanının sınıfına atanmasıdır.

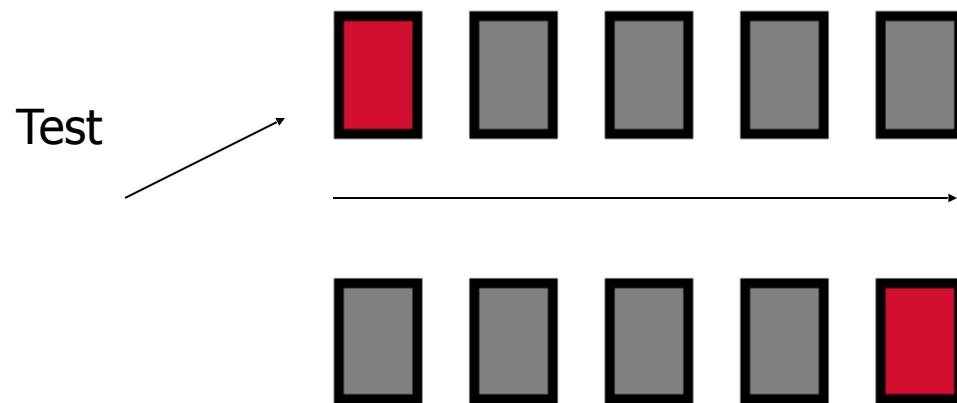


K-Fold Cross Validation (Çapraz Geçerleme)

Tüm dataseti eşit boyutlu N gruba böl



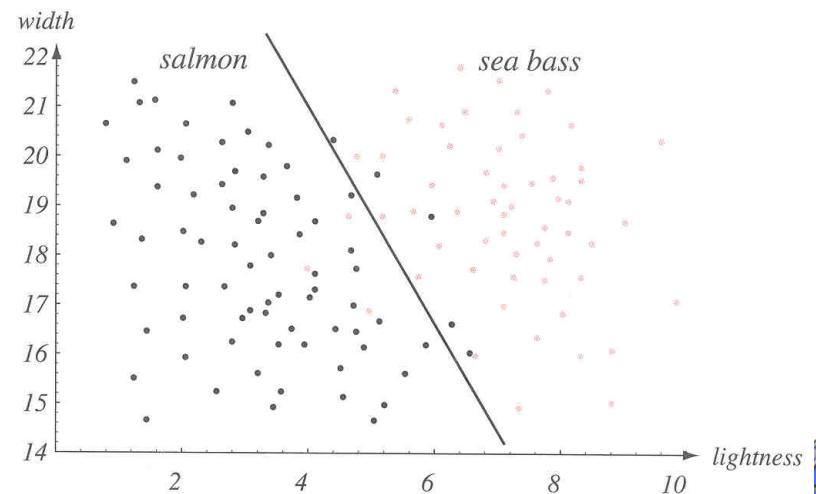
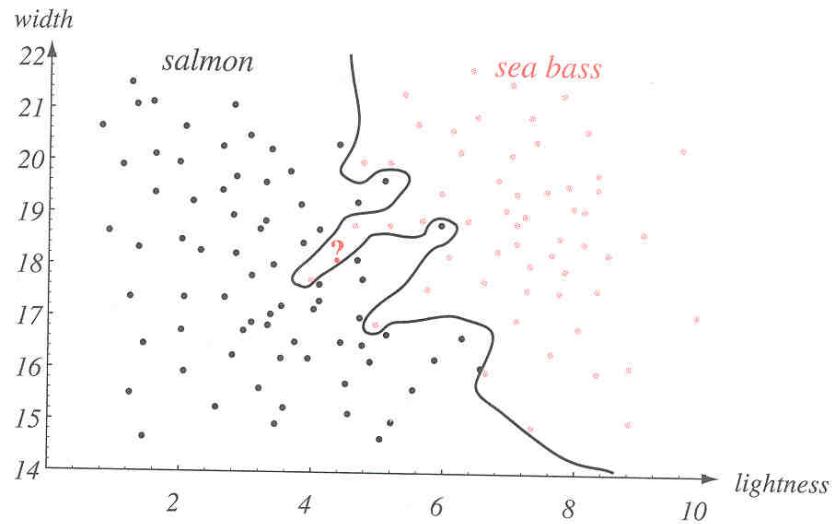
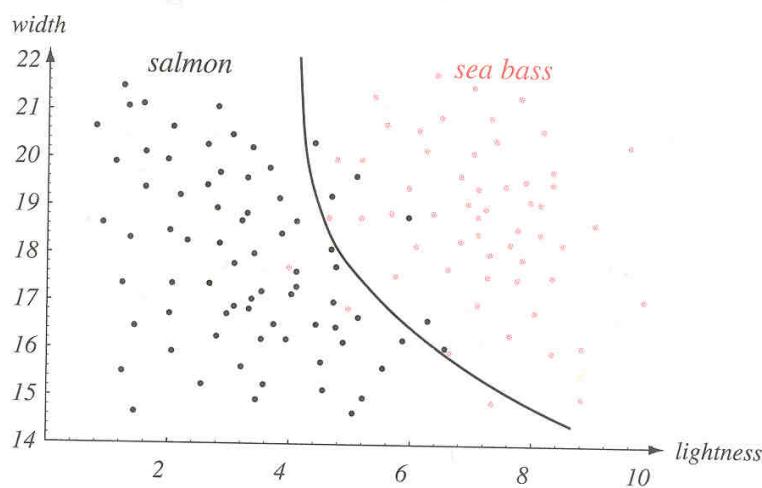
Bir grubu test için geriye kalanların hepsini eğitim için kullan



N defa tekrar et

Sınıflandırma Metotları- Sonuç

Farklı sınıflandırma metotlarının üretikleri karar sınırları



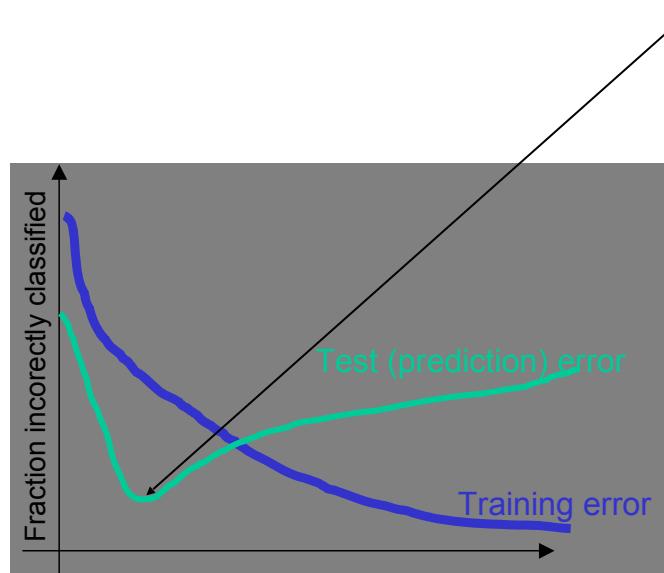
Sınıflandırma Metotları- Sonuç

- Neden bu kadar çok algoritma var ?
- Ne zaman hangisini seçeceğiz ?

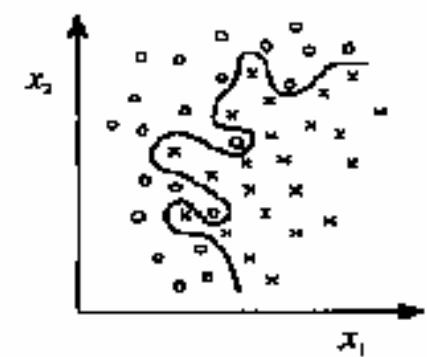
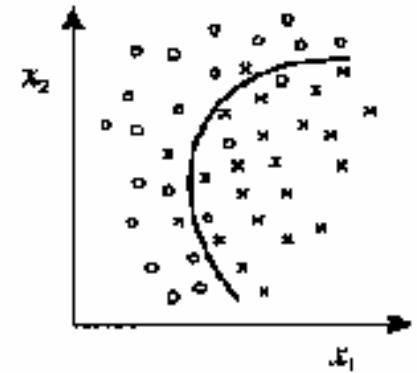
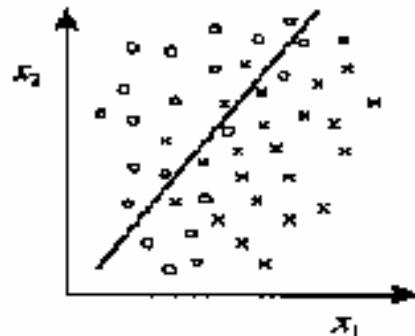
dataset	amlall	ann	bi75ds3	derma	gkanser	Hava
Özellik sayısı	7129	21	470	34	30	34
Sınıf sayısı	2	3	9	6	2	2
Örnek sayısı	72	3772	315	286	456	281
NB	97,14	95,55	68,49	77,97	94,29	89,31
SVM	92,86	93,74	62,11	79,37	96,26	86,48
1NN	94,29	93,4	63,19	76,26	96,26	89,72
C45	83,39	99,58	65,01	75,2	93,62	91,82
RF	95,71	99,5	72	76,96	95,38	95,02

Modelin karmaşıklığı arttığında eğitim kümelerindeki hata düşerken test kümelerindeki hata yükselir.

Her veri kümesi için optimum nokta (optimum karmaşıklık) farklıdır.



Model karmaşıklığı



AKİŞ

- Makine Öğrenmesi Nedir ?
- Günlük Hayatımızdaki Uygulamaları
- Verilerin Sayısallaştırılması
- Özellik Belirleme
 - Özellik Seçim Metotları
 - Bilgi Kazancı (Information Gain-IG)
 - Sinyal-Gürültüye Oranı: (S/N ratio)
 - Alt küme seçiciler (Wrappers)
 - Yeni Özelliklerin Çıkarımı
 - Temel Bileşen Analizi (Principal Component Analysis)
 - Doğrusal Ayırteden Analizi (Linear Discriminant Analysis)
- Sınıflandırma Metotları
 - Doğrusal Regresyon
 - SVM (Support Vector Machine)
 - Karar Ağaçları (Decision Trees)
 - Yapay Sinir Ağları
 - En Yakın K Komşu Algoritması (k - Nearest Neighbor)
 - Öğrenmeli Vektör Kuantalama (Learning Vector Quantization)
- **Kümeleme Algoritmaları**
 - K-Ortalama (K-Means)
 - Kendi Kendini Düzenleyen Haritalar (Self Organizing Map -SOM)

Kümeleme Algoritmaları

- Kümeleme algoritmaları eğiticisiz öğrenme metotlarıdır.
- Örneklerde ait sınıf bilgisini kullanmazlar.
- Temelde verileri en iyi temsil edecek vektörleri bulmaya çalışırlar.
- Verileri temsil eden vektörler bulunduktan sonra artık tüm veriler bu yeni vektörlerle kodlanabilirler ve farklı bilgi sayısı azalır.
- Bu nedenle birçok sıkıştırma algoritmasının temelinde kümeleme algoritmaları yer almaktadır.

Kümeleme Algoritmaları

Elimizde tek boyutlu 10 örnek içeren bir verimiz olsun.

12 15 13 87 4 5 9 67 1 2

Bu 10 farklı veriyi 3 farklı veri ile temsil etmek istersek

12 12 12 77 3 3 3 77 3 3 şeklinde ifade ederiz.

Gerçek değerler ile temsil edilen değerler arasındaki farkı minimum yapmaya çalışır.

Yukarıdaki örnek için 3 küme oluşmuştur.

12-15-13 örnekleri 1. kümede

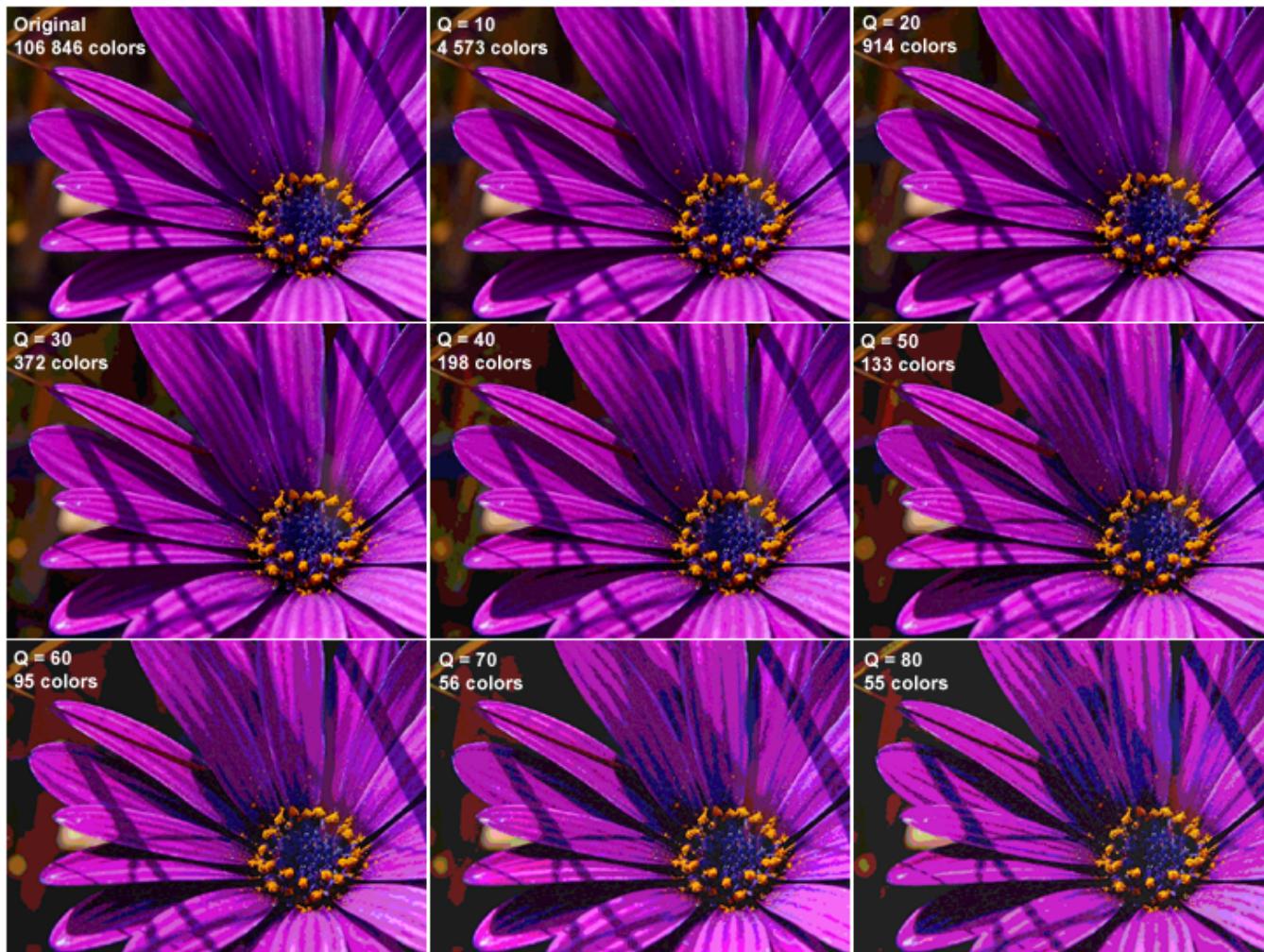
87-67 örnekleri 2. kümede

4-5-1-2-9 örnekleri 3. kümede yer almaktadır.



Renk Kumeleme

Quantization process



Resimdeki 106846 farklı renk sayısı 55 renge indirilmiştir

Resim Kümeleme



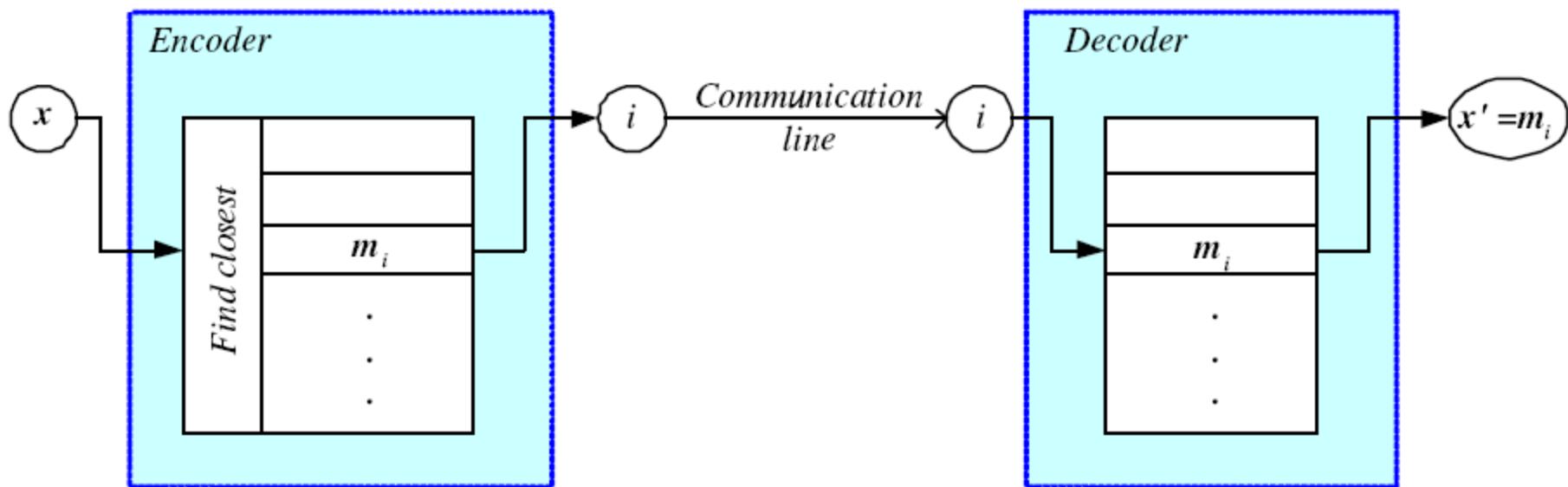
10*10 luk blokları ifade eden vektörler kümelenmiş

Nasıl Kullanılır ?

Bulunan (renkleri yada blokları temsil eden)

küme merkezlerinden bir kod kitabı (codebook) oluşturulur. Bu kitap

her iki merkeze verilir. Vektörlerin kendileri yerine sadece indisler kullanılır. İndisin maksimum büyüğünü kodlanması için gereken bit sayısını arttırır. Bu yüzden farklı vektör sayısının az olması istenir.

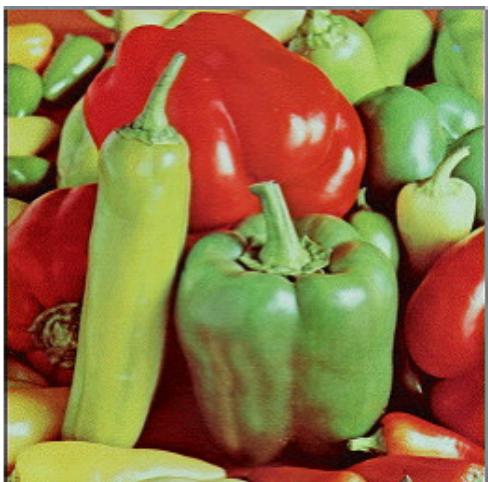


Kümelemede yaygın olarak kullanılan iki yöntem vardır.

- K-Means
- SOM (Self Organizing Map)

K-Means

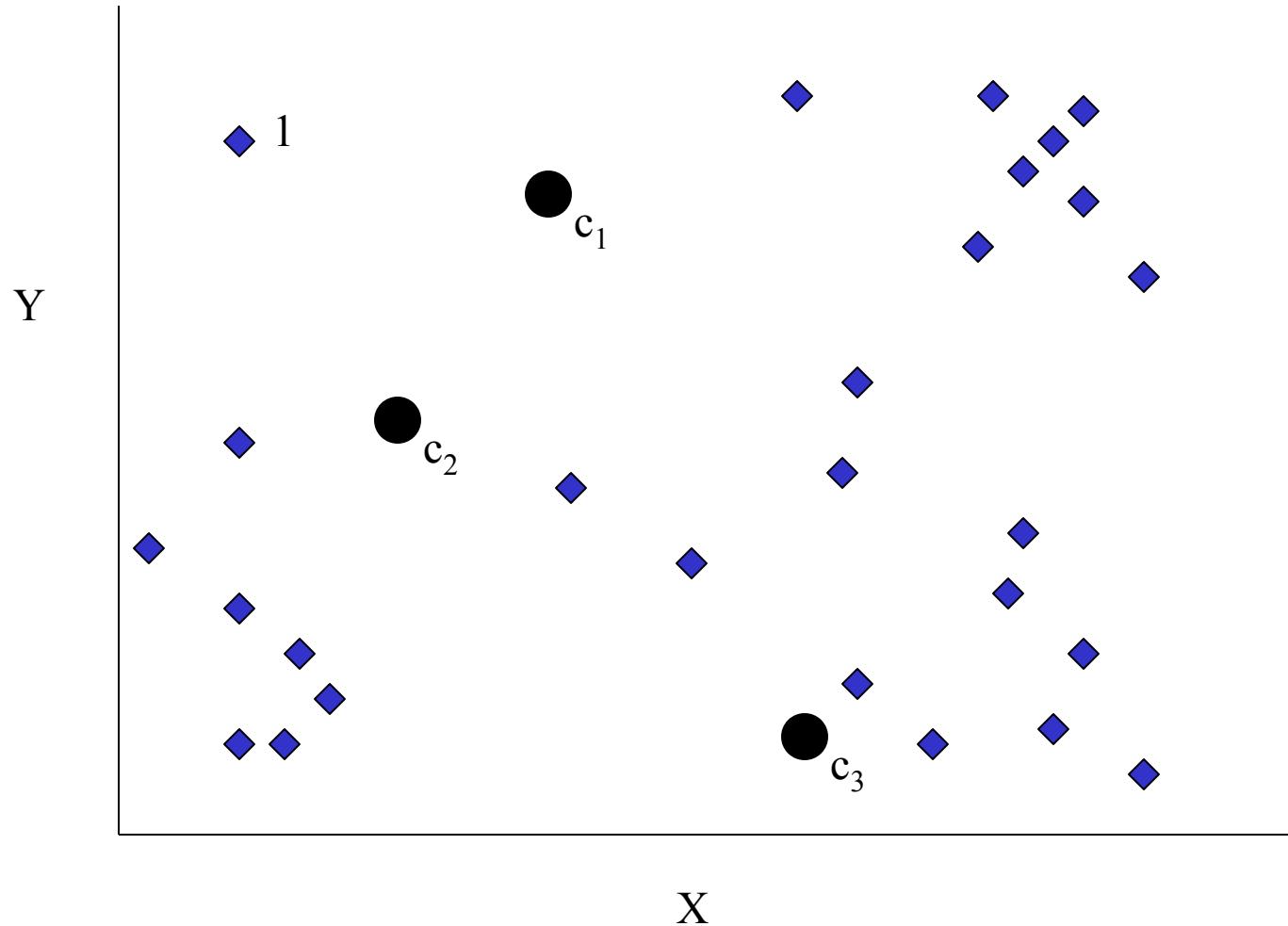
- Kümeleme algoritmalarının en basitidir. Veriyi en iyi ifade edecek K adet vektör bulmaya çalışır. K sayısı kullanıcı tarafından verilir. Nümerik değerler için çalışır.
 - i adet merkez belirlemek için :
 - Rasgele K adet küme merkezi atanır (C_1, C_2, \dots, C_k)
 - Her örnek en yakınındaki merkezin kümese atılır
 - C_i 'ler tekrar hesaplanır (her kümedeki örnegin ortalaması alınır)
 - C_i 'lerde değişiklik olmuş ise 2. ve 3. adımlar tekrar edilir. Bu işleme küme değiştiren örnek kalmayincaya kadar devam edilir, aksi takdirde işlem sonlandırılır.



Solda 256 renkle ifade edilen resim, sağda da K-Means kullanılarak 16 renge indirilmiş resim görülmektedir.

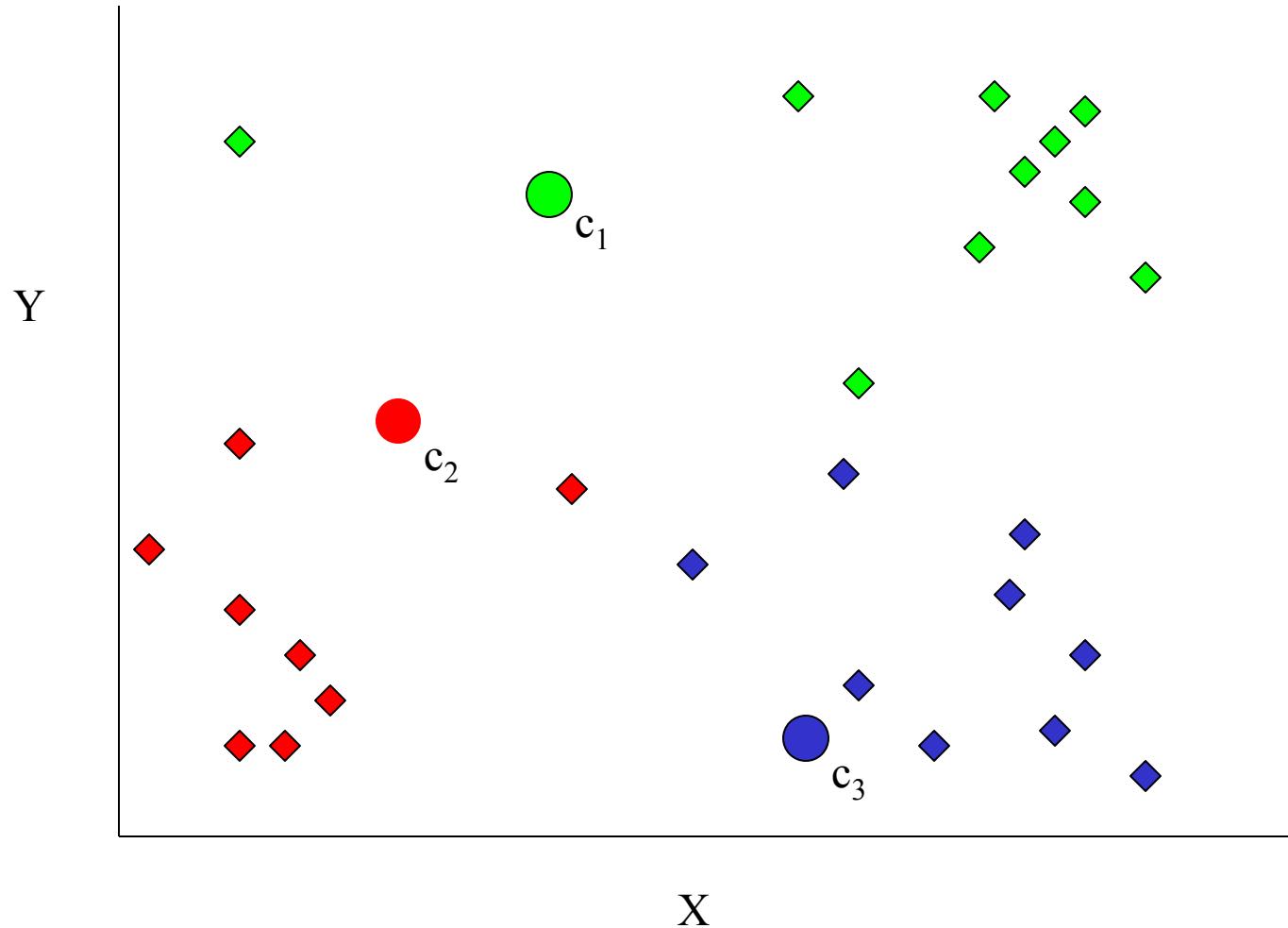
K-means örnek adım 1

Rasgele
3 küme
merkezi
ata.



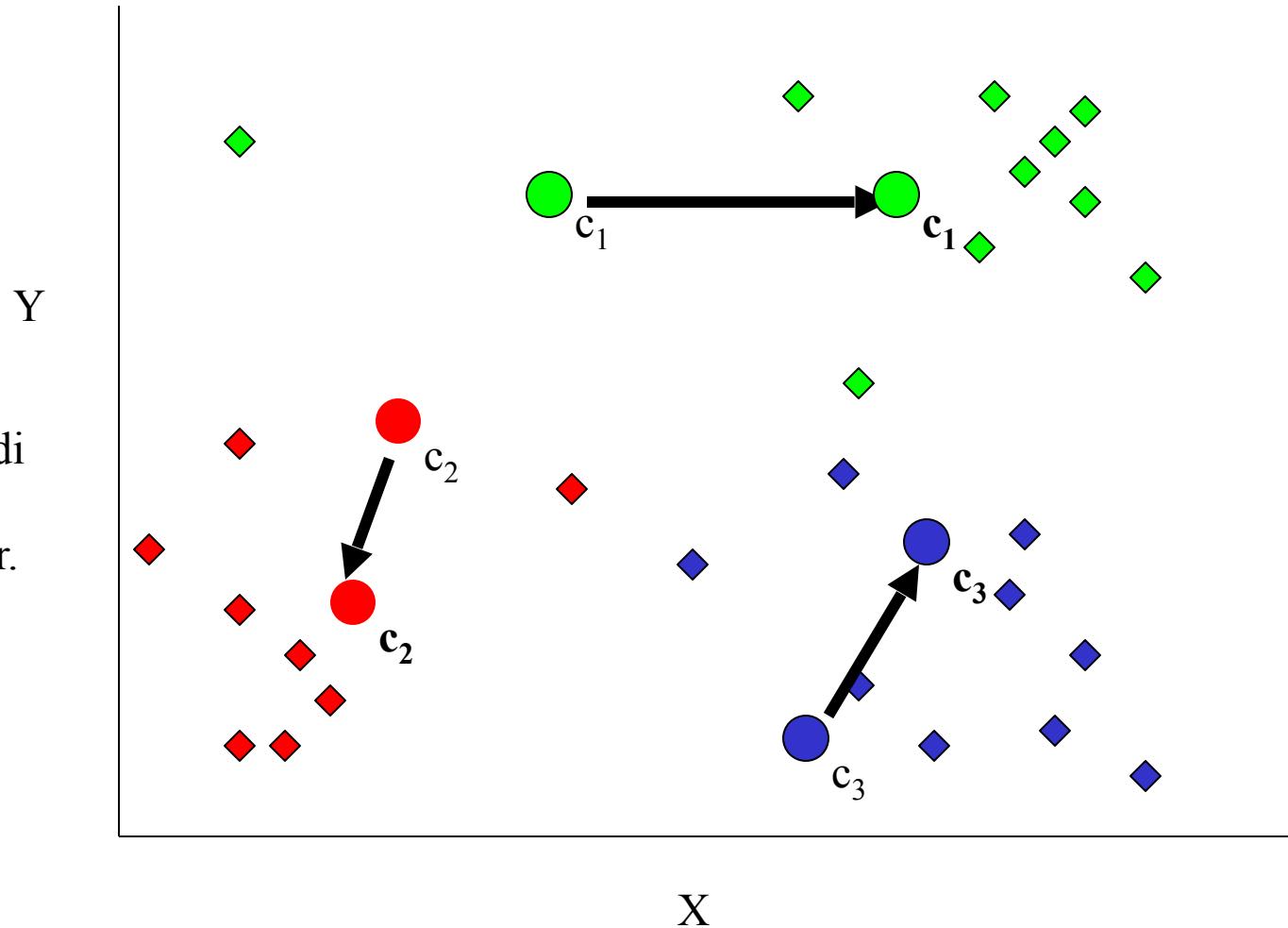
K-means örnek adım 2

Her örnek
en yakındakı
merkezin
kümesine atanır.



K-means örnek adım 3

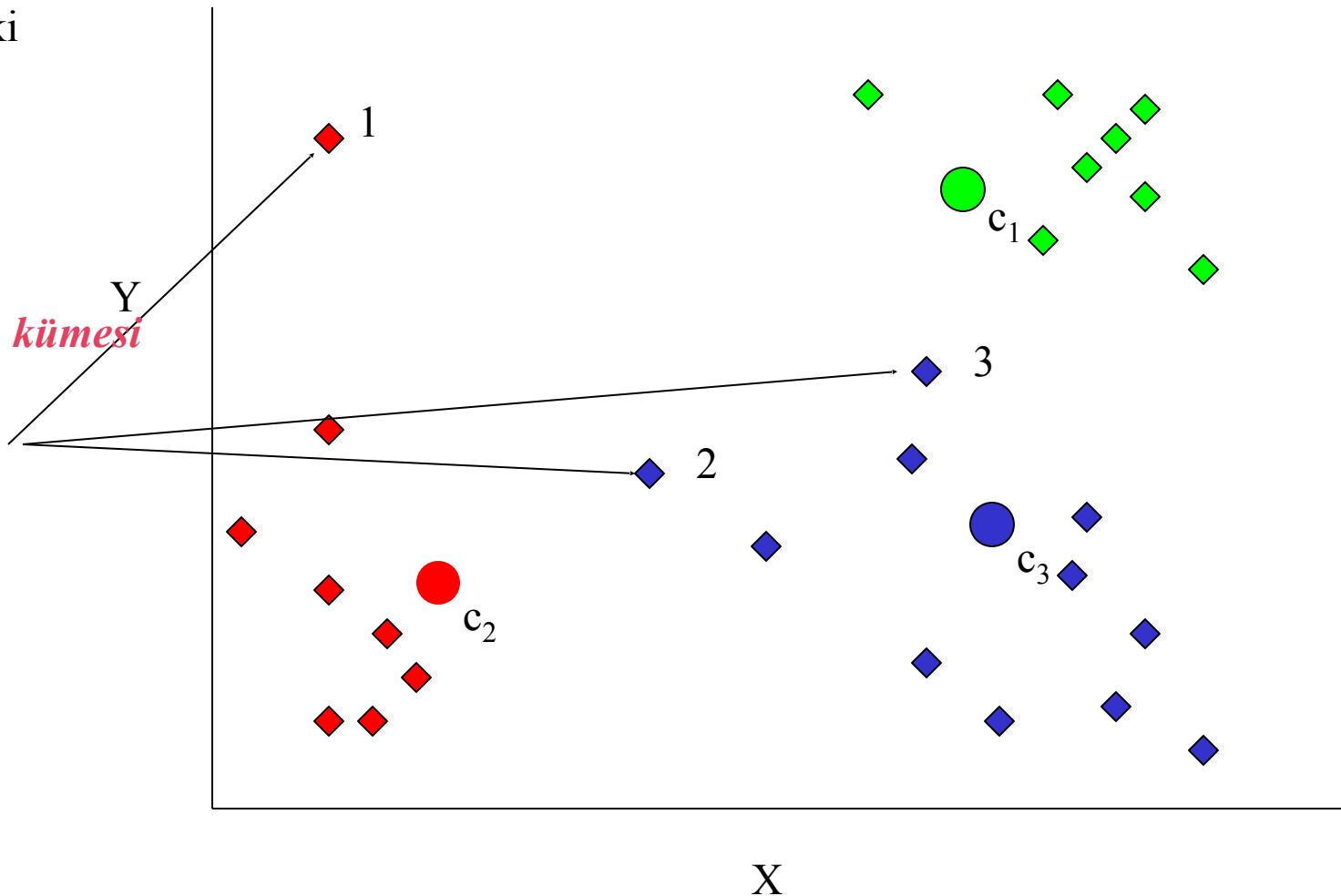
Merkezleri kendi
kümelerinin
merkezine götür.



K-means örnek adım 4

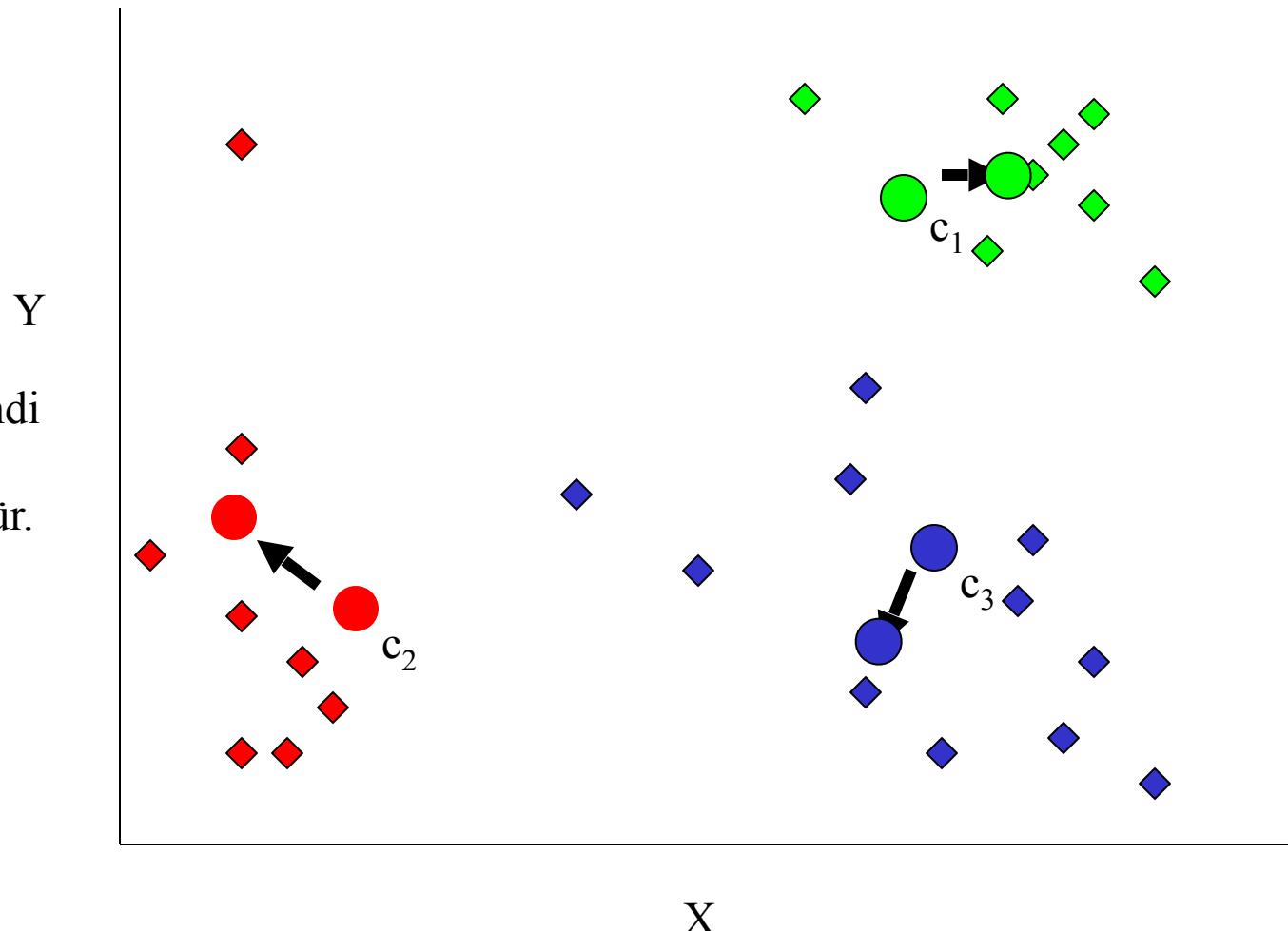
Her örneği
yeniden en
yakınındaki
merkezin
kümesine
ata.

*Q: Hangi
örneklerin kümesi
değişti?*



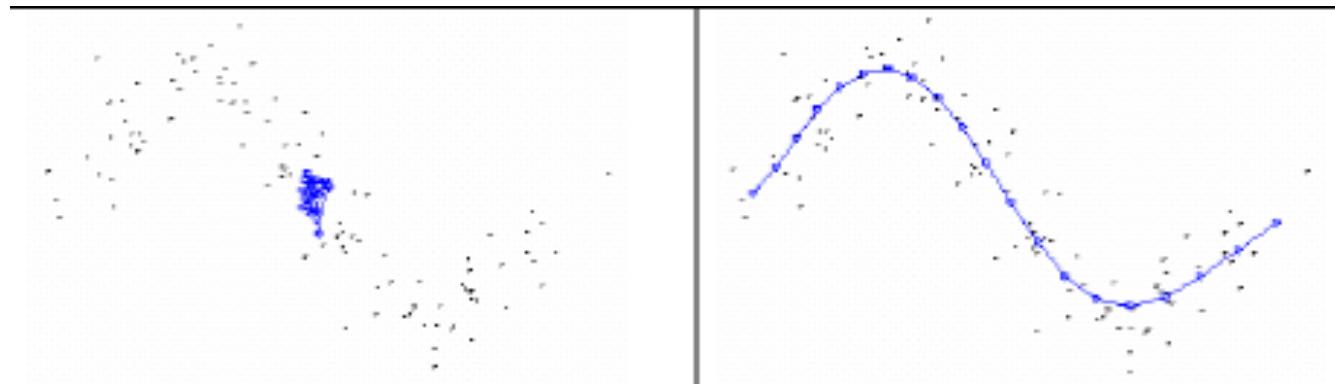
K-means örnek adım 5

Merkezleri kendi
kümelerinin
merkezine götür.



Kendi Kendini Düzenleyen Haritalar (SOM)

- K-Means algoritmasında merkez noktalar arasında herhangi bir ilişki yok iken SOM'da merkez noktalar 1 veya 2 boyutlu dizi içerisinde yer alırlar.
- K-Means algoritmasında sadece kazanan merkez güncellenirken, SOM 'da bütün merkezler kazanan nöronun komşuluklarına göre güncellenirler. Yakın komşular uzak komşulara göre daha fazla hareket eder.



SOM merkezleri 1 boyutlu bir dizide birbirlerine komşudur, başlangıçta rasgele atandıkları için yiğilma mevcuttur ancak eğitim tamamlandıktan sonra SOM merkezleri düzgün dağılmıştır.

Sonuç

Makineler insanlığın işgücüne sağladıkları katkıyı, makine öğrenmesi metotları sayesinde insanlığın beyin gücüne de sağlamaya başlamışlardır



Bir gün bilgisayarlar bizi anarlarsa ?

Ve bütün bunları mükemmel bir şekilde yaparlarsa Nasıl bir dünya bizi bekler

- Bir sürü işsiz bilgisayar mühendisi ☺
- Bir sürü işsiz insan
- ???



Kaynaklar

- Alpaydın E. (2004) "Introduction to Machine Learning", The MIT Press, 3-6
- <http://www.autonlab.org/tutorials/infogain11.pdf>
- http://www.kdnuggets.com/dmcourse/data_mining_course/assignments/assignment-4.html
- http://pespmc1.vub.ac.be/asc/SENSIT_ANALY.html
- http://csnet.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf
- http://www.cavs.msstate.edu/hse/ies/publications/reports/isip_internal/1998/linear_discrim_analysis/lda_theory.pdf
- <http://www.kernel-machines.org>
- T.Kohonen," Self-Organization and associative Memory",3d ed, 1989, Berlin :Springer-Verlag.
- <http://www.willamette.edu/~gorr/classes/cs449/Classification/perceptron.html>
- O. T. Yıldız, E. Alpaydın, Univariate and Multivariate Decision Trees, Tainn 2000
- <http://www.ph.tn.tudelft.nl/PHDTheses/AHoekstra/html/node45.html>
- <http://mathworld.wolfram.com/K-MeansClusteringAlgorithm.html>
-

Weka



Copyright: Martin Kramer (mkramer@wxs.nl)

YILDIZ TEKNİK ÜNİVERSİTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

